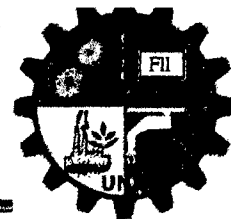




**UNIVERSIDAD NACIONAL DE PIURA**  
**FACULTAD DE INGENIERÍA INDUSTRIAL**  
**DECANATO**



**ACTA DE SUSTENTACIÓN DE TESIS**

Los Miembros del Jurado Calificador de la Tesis denominada: «**DESARROLLO DE UN SISTEMA PARA LA GENERACIÓN DE LA PROGRAMACIÓN ACADÉMICA DE LA FACULTAD DE INGENIERÍA INDUSTRIAL BASADO EN ALGORÍTMOS GENÉTICOS**», presentado por la señorita **VILCHEZ PALACIOS MARY CARMEN**, Bachiller de la Escuela Profesional en **INGENIERÍA INFORMÁTICA**, asesorada por el Dr. **MIGUEL JIMÉNEZ CARRIÓN**, Reunidos para la sustentación de ésta y luego de escuchar su exposición y las respuestas a las preguntas formuladas, la declaran:



Con el Calificativo:

*Aprobado*  
*Muy bueno*

En consecuencia la sustentante se encuentra **apta** para recibir el título profesional de **INGENIERO INFORMÁTICO** conforme a Ley.

Piura, 28 de Marzo del 2016

**ING. CARMEN ZULEMA QUITO RODRÍGUEZ.**  
PRESIDENTE - JURADO CALIFICADOR

**ING. PABLO DELGADO DÍAZ.**  
VOCAL - JURADO CALIFICADOR

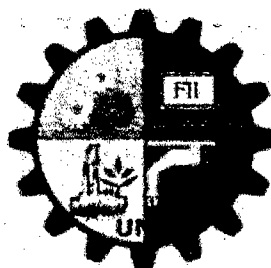
**ING. CARLOS ENRIQUE COELLO OBALLE.**  
SECRETARIO - JURADO CALIFICADOR

**UNIVERSIDAD NACIONAL DE PIURA**

**FACULTAD DE INGENIERIA INDUSTRIAL**

**ESCUELA PROFESIONAL DE INGENIERIA**

**INFORMATICA**



**DESARROLLO DE UN SISTEMA PARA LA GENERACION DE LA  
PROGRAMACION ACADÈMICA DE LA FACULTAD DE  
INGENIERIA INDUSTRIAL BASADO EN ALGRITMOS**

**GENÈTICOS**

**TESIS PARA OPTAR EL TITULO DE  
INGENIERO INFORMATICO**

**Presentado Por:**

**Bach. Mary Carmen Vilchez Palacios**

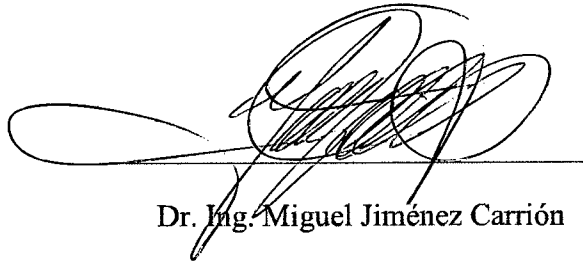
**PIURA - PERÚ**

**2016**

## REGISTRO DE FIRMAS


Tesis presentada como requisito para optar el título de Ingeniero Informático

Asesor:



Dr. Ing. Miguel Jiménez Carrión

Tesista:



Bach. Mary Carmen Vilchez Palacios

## **DEDICATORIA**

Esta investigación está dedicada a mi familia y a mis amigos por todo su cariño y apoyo incondicional.

## **AGRADECIMIENTOS**

Agradezco a Dios por permitirme disfrutar de la vida, a mi familia por brindarme su apoyo y a mi asesor Dr. Ing. Miguel Jiménez Carrión por su ayuda incondicional durante el desarrollo de la investigación.

## RESUMEN

Con la investigación realizada se ha logrado el desarrollo de un sistema para generar la programación académica en la Facultad de Ingeniería Industrial, además se ha planteado un modelo matemático y modelado las restricciones de acuerdo al análisis previamente realizado al proceso. También se ha diseñado el algoritmo genético y planteado la función de calidad teniendo en cuenta que dicha configuración permita trabajar al sistema sin afectar su performance, el problema de la elaboración de la programación académica es dividido en tres etapas diferentes y en cada una de ellas se asigna un recurso diferente, esto es beneficioso para el sistema dado que la división del problema en etapas permite disminuir tiempos debido a que solo se tiene que evaluar las restricciones relacionadas a cada etapa.

La secuencia para la asignación de los recursos es diferente, siendo la etapa de asignación de docentes aquella que no necesita del algoritmo genético; requiriendo para ello un algoritmo de asignación secuencial debido a sus características de asignación; las pruebas realizadas sobre los métodos de los operadores genéticos permitieron determinar cuáles son aquellos métodos que presentan un mejor comportamiento para el desarrollo del problema, siendo estos diferentes en las dos etapas. Además las pruebas permitieron observar que la variable número de generaciones no tiene influencia estadística de ninguna otra.

El sistema construido permite generar una programación académica sin violación de restricciones obligatorias, por lo que se afirma que se logró automatizar un proceso en la Facultad de Ingeniería Industrial y que se cumplió con el objetivo planteado. Además el sistema provee de soluciones en un tiempo menor al empleado por el Secretario Académico.

**Palabras Claves:** *Timetabling*, Algoritmo genéticos.

## ABSTRACT

The research has succeeded in developing a system for generating the academic program at the School of Industrial Engineering, also it has proposed a mathematical model and modeling constraints according to the analysis previously made to the process. Has also been designed genetic algorithm and proposed the quality function considering that this configuration allows to work the system without affecting its performance, the problem of the development of the academic program is divided into three different stages and in each of them assigns a different resource, this is beneficial to the system because the division of the problem in stages allows lower times because only have to evaluate related constraints to each stage.

The sequence for resource assignment is different, being the step of assigning teachers that do not need the genetic algorithm; requiring only sequential allocation algorithm due to its characteristics assignment; tests on methods of genetic operators helped to identify those methods that present a better behavior for the development of the problem is, these being different in the two stages. Furthermore, the evidence suggested that the variable number of generations is not statistically any other influence.

The built system can generate an academic program without violation of required constraint, which states that managed to automate a process at the Faculty of Industrial Engineering and met its objective. The system also provides solutions to a lesser to that used by the Academic Secretary time.

**Keywords:** *Timetabling, Genetic algorithm.*

## TABLA DE CONTENIDO

RESUMEN .....	v
ABSTRACT .....	vi
TABLA DE CONTENIDO .....	vii
ÍNDICE DE FIGURAS .....	xi
ÍNDICE DE TABLAS .....	xiv
ÍNDICE DE CUADROS .....	xv
INTRODUCCIÓN .....	1
CAPÍTULO 1: EL PROBLEMA DE INVESTIGACIÓN.....	3
1.1 Descripción de la realidad problemática .....	3
1.2 Formulación del problema.....	4
1.3 Justificación.....	5
1.4 Importancia.....	5
1.5 Objetivos .....	6
1.5.1 General .....	6
1.5.2 Específicos.....	6
1.6 Hipótesis.....	6
1.6.1 Identificación y operacionalización de la variable .....	7
CAPÍTULO 2: MARCO TEÓRICO.....	11
2.1 Bases teórico científicas .....	11
2.2 Antecedentes de la investigación .....	19
CAPÍTULO 3: DISEÑO DEL ALGORITMO GENÉTICO .....	20
3.1 Análisis del problema.....	20
3.1.1 Formulación del modelo matemático .....	22
3.2 Diseño del cromosoma.....	29
3.3 Implementación de la función de aptitud .....	31



3.4	Selección de operadores .....	33
3.4.1	Selección .....	33
3.4.2	Cruce .....	33
3.4.3	Mutación.....	34
3.4.4	Zap.....	34
3.5	Desarrollo del prototipo.....	34
3.5.1	Determinación de los parámetros de entrada.....	34
3.5.2	Diagrama de flujo del AGs.....	36
3.6	Implementación del prototipo .....	37
3.6.1	Asignación de docentes.....	38
3.6.2	Asignación de grupos .....	39
3.6.3	Asignación de aulas.....	48
CAPÍTULO 4: CONCEPCIÓN, ANÁLISIS Y DISEÑO DEL SISTEMA.....		55
4.1	Casos de uso del sistema .....	55
4.1.1	Descripción de casos de uso .....	55
4.1.1.1	Caso de uso “Ingresar al sistema” .....	56
4.1.1.2	Caso de uso “Configurar disponibilidad de aula” .....	57
4.1.1.3	Caso de uso “Configurar asignación docente”.....	58
4.1.1.4	Caso de uso “Configurar disponibilidad de horarios”.....	59
4.1.1.5	Caso de uso “Configurar preferencia de dictado de cursos” .....	60
4.1.1.6	Caso de uso “Configurar programación de cursos” .....	62
4.1.1.7	Caso de uso “Configurar horario” .....	64
4.1.1.8	Caso de uso “Configurar parámetros”.....	65
4.1.1.9	Caso de uso “Generar programación académica” .....	66
4.2.1	Diagrama de secuencia “Ingreso al sistema” .....	68
4.2.2	Diagrama de secuencia “Configurar disponibilidad de aula” .....	69

4.2.3	Diagrama de secuencia “Configurar asignación docente” .....	69
4.2.4	Diagrama de secuencia “Configurar disponibilidad de horarios” .....	70
4.2.5	Diagrama de secuencia “Configurar preferencia de dictado de cursos” .....	71
4.2.6	Diagrama de secuencia “Configurar programación de cursos” .....	72
4.2.7	Diagrama de secuencia “Configurar horario” .....	73
4.2.8	Diagrama de secuencia “Configurar parámetros” .....	73
4.2.9	Diagrama de secuencia “Generar programación académica” .....	74
4.3	Diagrama entidad relación.....	75
4.4	Diagrama de componentes .....	76
4.5	Diagrama de despliegue .....	76
4.6	Interfaces del sistema .....	77
4.6.1	Interfaz “Acceso de usuario” .....	77
4.6.2	Interfaz “Configurar disponibilidad de aula” .....	77
4.6.3	Interfaz “Configuración de docentes” .....	78
4.6.4	Interfaz “Configurar programación de cursos” .....	79
4.6.5	Interfaz “Configurar horario” .....	79
4.6.6	Interfaz “Configurar parámetros” .....	80
4.6.7	Interfaz “Generar programación académica” .....	80
CAPITULO 5: PRUEBAS Y CALIBRACIÓN DEL ALGORITMO.....		81
5.1	Pruebas sobre los métodos de los operadores.....	81
5.1.1	Pruebas sobre los proceso de selección .....	81
5.1.2	Pruebas sobre el proceso de cruce .....	82
5.1.3	Pruebas sobre el proceso de mutación.....	83
5.2	Pruebas sobre los parámetros .....	83
5.2.1	Prueba para el tamaño de población.....	84
5.2.2	Prueba para el porcentaje de cruce .....	84

5.2.3 Prueba para el porcentaje de mutación .....	85
CAPÍTULO 6: ANÁLISIS Y DISCUSIÓN DE RESULTADOS .....	86
CONCLUSIONES .....	91
RECOMENDACIONES.....	92
BIBLIOGRAFÍA .....	93
ANEXOS .....	95

## ÍNDICE DE FIGURAS

Figura 3.1 Diagrama de procesos para la elaboración de la P.A. ....	21
Figura 3.2 Unidad de asignación .....	30
Figura 3.3 Recursos asignables.....	30
Figura 3.4 Diseño del cromosoma .....	31
Figura 3.5 Diagrama de flujo del AGs.....	36
Figura 3.6 Pasos para generar un individuo – Asignación de grupos .....	41
Figura 3.7 Población inicial - Asignación de grupos .....	42
Figura 3.8 Evaluación de RO4.....	42
Figura 3.9 Evaluación de RO5.....	43
Figura 3.10 Evaluación de RO6.....	44
Figura 3.11 Selección por torneo.....	45
Figura 3.12 Cruce de 2 puntos .....	45
Figura 3.13 Población de hijos - Asignación de grupos .....	46
Figura 3.14 Mutación de un individuo - Asignación de grupos .....	47
Figura 3.15 Pasos para generar un individuo – Asignación de aulas.....	49
Figura 3.16 Población inicial - Asignación de aulas.....	50
Figura 3.17 Evaluación de RO7.....	50
Figura 3.18 Cruce Uniforme.....	51
Figura 3.19 Población de hijos - Asignación de aulas .....	52
Figura 3.20 Mutación de 1 individuo - Asignación de aulas .....	52
Figura 4.1 Diagrama de casos de uso .....	55

Figura 4.2 Diagrama de secuencia “Ingreso al sistema” .....	68
Figura 4.3 Diagrama de secuencia “Configurar disponibilidad de aula” .....	69
Figura 4.4 Diagrama de secuencia “Configurar asignación docente” .....	69
Figura 4.5 Diagrama de secuencia “Configurar disponibilidad de horarios” .....	70
Figura 4.6 Diagrama de secuencia “Configurar preferencia de dictado de cursos” .....	71
Figura 4.7 Diagrama de secuencia “Configurar programación de cursos” .....	72
Figura 4.8 Diagrama de secuencia “Configurar horario” .....	73
Figura 4.9 Diagrama de secuencia “Configurar parámetros” .....	73
Figura 4.10 Diagrama de secuencia “Generar programación académica” .....	74
Figura 4.11 Diagrama entidad relación.....	75
Figura 4.12 Diagrama de componentes .....	76
Figura 4.13 Diagrama de despliegue .....	76
Figura 4.14 Interfaz "Acceso de usuario" .....	77
Figura 4.15 Interfaz "Menú principal" .....	77
Figura 4.16 Interfaz "Configurar disponibilidad de aula" .....	78
Figura 4.17 Interfaz "Configuración de asignación docente" .....	78
Figura 4.18 Interfaz "Configurar programación de cursos" .....	79
Figura 4.19 Interfaz "Configurar horario" .....	79
Figura 4.20 Interfaz "Configurar parámetros" .....	80
Figura 4.21 Interfaz "Generar programación académica" .....	80
Figura 5.1 Métodos de selección vs Número de generaciones .....	82
Figura 5.2 Métodos de cruce vs Número de generaciones .....	82
Figura 5.3 Métodos de mutación vs Número de generaciones .....	83

Figura 5.4 Tamaño de la población vs. Número de generaciones .....	84
Figura 5.5 Porcentaje de cruza vs Número de generaciones.....	84
Figura 5.6 Porcentaje de mutación vs Número de generaciones .....	85

## ÍNDICE DE TABLAS

Tabla 3.1 Asignación de docentes .....	39
Tabla 3.2 Asignación de grupos .....	48
Tabla 3.3 Asignación de aulas .....	53
Tabla 3.4 Ejemplo de programación de clases.....	54
Tabla 6.1 Resultados de las generaciones.....	88

## ÍNDICE DE CUADROS

Cuadro 6.1 Análisis de varianza para “Número de generaciones” .....	86
---	----



## INTRODUCCIÓN

Los avances tecnológicos son cada vez mayores y cada día se vuelve más importante para las distintas organizaciones y/o empresas automatizar sus procesos, es necesario que se busquen métodos que ayuden a dar soluciones a los diferentes procesos en un menor tiempo y de manera óptima que realizándolos de la forma tradicional.

Las instituciones que brindan el servicio de educación como la Universidad Nacional de Piura tienen al inicio de cada semestre académico la tarea de elaborar la programación de cursos a dictar correspondiente a las diferentes facultades. En la Facultad de Ingeniería Industrial (FII) la elaboración de esta programación está a cargo del Secretario Académico siendo este el encargado de asignar las aulas y grupos a los diferentes cursos, enviando después esta lista a los correspondientes departamentos académicos para la asignación de docentes.

Considerando la cantidad de cursos que se necesitan abrir, las aulas disponibles y los diferentes grupos de horarios que se tienen por semestre académico se consideran diferentes opciones de asignación teniendo que ser escogida la opción que cumpla con ciertas restricciones para ser considerada óptima. La cantidad de datos con los que se cuenta, hacen que esta labor requiera de un tiempo considerable para su realización y el cumplimiento de ciertas restricciones limita el espacio de búsqueda, es por ello que el proyecto de investigación es planteado con el fin de encontrar un modelo que represente eficientemente el problema y además proponer un sistema para elaborar la programación académica utilizando algoritmos genéticos con un enfoque dirigido a minimizar considerablemente el tiempo dedicado a realizar la programación de cursos y cuyo desarrollo del sistema implica un avance tecnológico debido a que se automatizará un proceso de la FII. En el escenario actual el Secretario Académico requiere un día para realizar la programación académica y aun así no garantiza que este libre de violación de restricciones, siendo la más frecuente el cruce de clases en una misma aula y un mismo grupo.

El desarrollo de este trabajo está dividido en 6 capítulos, en el capítulo uno se presenta un enfoque general del problema de asignación de recursos en un espacio de tiempo específico considerado como problema de tipo *timetabling*, además se describe la importancia, justificación y objetivos del proyecto. El capítulo dos está dedicado a presentar una descripción de los algoritmos genéticos, sus principales características, sus operadores

de evolución y la configuración paramétrica necesaria para poder trabajar con ellos. En el capítulo tres abordamos el problema con un enfoque específico basándonos en las características del proceso que se sigue en la FII con la finalidad de identificar los datos involucrados, las variables de decisión, plantear la función objetivo y las diferentes restricciones para representar el problema en un modelo matemático, el problema tuvo que ser abordado en etapas diferentes con la finalidad de disminuir su complejidad, observándose que en dos de ellas se ha implementado el principio evolutivo de los algoritmos genéticos. Las etapas fueron divididas de acuerdo a los recursos que son necesarios asignar (docentes, grupos y aulas), donde dos de ellas hicieron uso del proceso evolutivo, en este capítulo además se presenta el diseño del cromosoma, la implementación de la función de evaluación, la configuración de los diferentes parámetros y la implementación del algoritmo genético. En el capítulo cuatro se presenta el análisis y diseño del sistema. En el capítulo cinco se describen las pruebas realizadas y calibración del algoritmo. En el capítulo seis se realiza un análisis de los resultados y por último se presentan las conclusiones y recomendaciones.

# **CAPÍTULO 1: EL PROBLEMA DE INVESTIGACIÓN**

## **1.1 DESCRIPCIÓN DE LA REALIDAD PROBLEMÁTICA**

En todas las instituciones de educación superior existe un proceso para generar la programación académica, este proceso permite asignar adecuadamente recursos tales como: docentes, aulas y cursos en espacios de tiempo específicos que en algunos casos están agrupados en horarios; estos recursos son limitados y en consecuencia las personas encargadas de esta asignación tienen la necesidad de distribuir eficientemente dichos recursos en beneficio de los estudiantes, además de la planificación de horarios de clase es necesario determinar qué cursos se deben abrir cuando estos están fuera de los ciclos académicos correspondientes al semestre dependiendo de las necesidades de los alumnos por llevarlos, tal situación descansa en la disponibilidad de los recursos, si es que aún existen.

La elaboración de dicha programación genera los horarios de clase correspondiente a los alumnos, teniendo ellos la posibilidad de escoger los cursos que deseen llevar dependiendo del ciclo y situación académica en la que se encuentren, la disposición de estos cursos ayudará al alumno a continuar satisfactoriamente su formación académica, es por ello que dicha programación académica debe culminar con resultados aceptables para evitar que los alumnos tengan problemas al momento de inscribirse en los cursos que le corresponden, uno de estos problemas puede ser el cruce de cursos correspondiente a un determinado semestre.

Al problema de asignación de recursos en espacios de tiempo específicos se le conoce como problema de horarios (*timetabling*). *Timetabling* es un problema muy complejo debido a los criterios y procesos que se deben realizar. En cuanto a los criterios de asignación se debe tener en cuenta el cumplimiento de restricciones que en caso de ser violadas generan horarios académicos erróneos, produciendo de esta manera insatisfacción por parte de alumnos y docentes; como por ejemplo, se debe evitar el cruce de cursos en una misma aula en un tiempo específico; el cumplimiento de estas restricciones hace que el espacio de soluciones sea acotado y tedioso de desarrollar, requiriendo de esta manera de un tiempo considerable para su realización. Los procesos que se consideran abarcan desde el momento en que se determina qué cursos se deben abrir, el horario de dictado de dichos cursos hasta la correcta asignación de docentes para dictar un curso.

Como se citó antes es necesario determinar cuántas secciones abrir en determinado curso, determinar qué docente con cierta preferencia en el dictado de los cursos y con cierta disponibilidad de horarios se asignará a la asignatura; además se dispone de aulas y grupos y la combinación de estos recursos darán lugar a la clase, a la cual se podrán escribir los alumnos.

En el caso particular, de la FII-UNP, el proceso para elaborar la programación académica se realiza prácticamente de forma manual, lo cual se torna subjetivo a la experiencia de quien asigna los recursos, asignándolos como mejor parecen ajustarse según su criterio; y como se mencionó antes, las restricciones obligatorias, no siempre se logran cumplir; como por ejemplo, realizar la asignación de un curso en la misma aula en una hora determinada. Este proceso lo inicia el Secretario Académico quién solicita a los Directores de cada Escuela la lista de cursos que se deben abrir en el siguiente periodo académico dependiendo de las necesidades de los alumnos, luego esta información es enviada al Secretario Académico quien será el encargado de asignar aulas a determinada asignatura en un grupo correspondiente, considerando la cantidad de datos involucrados en este proceso y las diferentes opciones de asignación que existen se considera una tarea engorrosa requiriendo de un tiempo considerable para su realización, además es posible que se deban hacer correcciones hasta lograr una solución aceptable; cuando dicha asignación es culminada es enviada a los diferentes Jefes de Departamento que se encargarán de asignar los docentes a cada curso, en la asignación de docentes también es necesario considerar la categoría de cada docente, número de horas mínimas de enseñanza, los cursos que enseñan de acuerdo a su experiencia y la preferencia por enseñar tales cursos en un determinado horario.

A pesar que la tecnología está en vanguardia aún existen instituciones que realizan sus procesos de manera manual, es por ello que es necesario que cuenten con herramientas que permitan optimizar dichos procesos para realizarlos más eficientemente, reduciendo así problemas de tiempo y la oportunidad de brindar un mejor servicio.

## **1.2 FORMULACIÓN DEL PROBLEMA**

La realización de esta investigación permitirá dar respuesta a la siguiente interrogante: ¿Un sistema basado en algoritmos genéticos reducirá el tiempo empleado en el proceso de elaboración de la programación académica?

### **1.3 JUSTIFICACIÓN**

El proceso de generación de horarios de clases es un problema por el que atraviesan las instituciones que brindan el servicio de educación, por lo general este proceso es realizado de forma manual y toma mucho tiempo para lograr una solución que algunas veces no concluye con resultados aceptables. Estas instituciones cuentan con recursos limitados (aulas, laboratorios, docentes) y necesitan realizar una mejor administración de estos recursos para así lograr satisfacción por parte de los alumnos y docentes.

Por estas razones se propone el desarrollo de este proyecto que tiene como finalidad minimizar el tiempo utilizado para la generación de los horarios de clase satisfaciendo de esta manera las necesidades de los usuarios; además la automatización de dicho proceso significa un avance tecnológico y el desarrollo del sistema permitirá una mejor gestión de los recursos disponibles.

### **1.4 IMPORTANCIA**

La importancia de esta investigación radica en proporcionar una herramienta informática basada en algoritmos genéticos que permita la generación de la programación académica, además dicha herramienta permitirá la asignación de recursos humanos y materiales en un tiempo menor al utilizado actualmente en la FII. Utilizar la herramienta implica un cambio en los procedimientos actualmente seguidos, no será necesario que los Directores de Escuela envíen la lista de cursos para los cuales se asignará un aula y horario sino que la lista de cursos se obtendrá directamente del sistema, la cual quedará como una opción a implementarse cuando se tenga acceso al sistema de gestión de base de datos de la Universidad Nacional de Piura, además se asignarán los recursos, tales como aula, grupo y docente a las diferentes clases disminuyendo de esta manera el tiempo empleado para la asignación de dichos recursos. También es importante porque permitirá encontrar una distribución adecuada de horarios para que los docentes dicten clases de acuerdo a su experiencia y disponibilidad de horarios, permitiendo de esta manera que las clases sean dictadas por docentes especializados teniendo los alumnos la oportunidad de sacar un mejor provecho de esto.

## **1.5 OBJETIVOS**

### **1.5.1 General**

Desarrollar un sistema basado en algoritmos genéticos que permita automatizar el proceso de elaboración de la programación académica en la FII.

### **1.5.2 Específicos**

- Analizar la realidad problemática en la elaboración de la programación académica para identificar las características de dicho proceso, definir los requerimientos del sistema y determinar la influencia del desarrollo del sistema sobre la realidad problemática.
- Definir las restricciones del modelo partiendo del análisis de la realidad problemática.
- Definir el modelo matemático identificando los elementos que deben componer dicho modelo y además determinar cuántas y cuales restricciones antes definidas deben cumplirse para que el algoritmo propuesto provea soluciones factibles.
- Diseñar el algoritmo genético que permita la generación de la programación académica de forma óptima basándose en las restricciones obtenidas.
- Diseñar las interfaces del sistema.
- Realizar las pruebas necesarias para garantizar que los resultados obtenidos cumplan con las condiciones antes previstas.

## **1.6 HIPÓTESIS**

El desarrollo de un sistema basado en algoritmos genéticos reducirá el tiempo de generación de la programación académica.

### **1.6.1 Identificación y operacionalización de la variable**

- Número de alumnos que existen por asignatura.
- Número de docentes disponibles a dictar cierta asignatura.
- Categoría del docente.
- Número de aulas.
- Horario de trabajo.
- Preferencia de los docentes para el dictado de las asignaturas.
- Sistema para la generación de la programación académica
- Reducir el tiempo de generación de la programación académica.

Indicador	Definición conceptual	Definición operacional	Unidad	Técnica	Fuente
Tiempo promedio para configurar la cantidad de aulas disponibles	Es la medida de tiempo que se requiere para configurar las aulas disponibles en el semestre académico.	$tpca = \sum_{i=1}^{ng} \frac{tf - ti}{NG}$ <p><i>tf</i>: Tiempo final del registro de la configuración de las aulas disponibles.  <i>ti</i>: Tiempo inicial del registro de la configuración de aulas disponibles.  <i>NG</i>: Número de generaciones.</p>	Minutos	Observación	Registros de tiempos tomados en situaciones simuladas.
Tiempo promedio para configurar los grupos disponibles	Es la medida de tiempo que se requiere para configurar los grupos disponibles en el semestre académico.	$tpcg = \sum_{i=1}^{ng} \frac{tf - ti}{NG}$ <p><i>tf</i>: Tiempo final del registro de la configuración de grupos disponibles.  <i>ti</i>: Tiempo inicial del registro de la configuración de grupos disponibles.  <i>NG</i>: Número de generaciones.</p>	Minutos	Observación	Registros de tiempos tomados en situaciones simuladas.
Tiempo promedio para configurar la asignación de docentes.	Es la medida de tiempo que se requiere para configurar qué docentes se tendrán en cuenta para la asignación de clases.	$tpcd = \sum_{i=1}^{ng} \frac{tf - ti}{NG}$	Minutos	Observación	Registros de tiempos tomados en situaciones simuladas.



			<p><i>tf</i>: Tiempo final del registro de la configuración de asignación de docentes.</p> <p><i>ti</i>: Tiempo inicial del registro de la configuración de asignación de docentes.</p> <p><i>NG</i>: Número de generaciones.</p>			
Tiempo promedio para configurar la información de los docentes.	Es la medida de tiempo que se requiere para configurar la información de los docentes (cursos que dictan, horarios).	$tpcif = \sum_{i=1}^{ng} \frac{tf - ti}{NG}$ <p><i>tf</i>: Tiempo final del registro de la información de los docentes.</p> <p><i>ti</i>: Tiempo inicial del registro de la información de los docentes.</p> <p><i>NG</i>: Número de generaciones.</p>	Minutos	Observación	Registros de tiempos tomados en situaciones simuladas.	
Tiempo promedio para configurar los valores de los parámetros.	Es la medida de tiempo que se requiere para configurar los valores de los parámetros para el funcionamiento del algoritmo genético.	$tpcp = \sum_{i=1}^{ng} \frac{tf - ti}{NG}$ <p><i>tf</i>: Tiempo final del registro de la configuración de valores de los parámetros.</p> <p><i>ti</i>: Tiempo inicial del registro de la configuración de valores de los parámetros.</p> <p><i>NG</i>: Número de generaciones.</p>	Minutos	Observación	Registros de tiempos tomados en situaciones simuladas.	

Tiempo promedio para configurar los cursos a programar por las diferentes escuelas.	Es la medida de tiempo que se requiere para configurar los cursos que se dictarán para las diferentes escuelas.	$tpcc = \sum_{i=1}^{ng} \frac{tf - ti}{NG}$ <p><i>tf</i>: Tiempo final del registro de la configuración de cursos a dictar.</p> <p><i>ti</i>: Tiempo inicial del registro de la configuración de cursos a dictar.</p> <p><i>NG</i>: Número de generaciones.</p>	Minutos	Observación	Registros de tiempos tomados en situaciones simuladas.
Tiempo promedio para generar la programación académica.	Es la medida de tiempo que se requiere para generar la programación académica.	$tpgp = \sum_{i=1}^{ng} \frac{tf - ti}{NG}$ <p><i>tf</i>: Tiempo final al generar la programación académica.</p> <p><i>ti</i>: Tiempo inicial al generar la programación académica.</p> <p><i>NG</i>: Número de generaciones.</p>	Minutos	Observación	Registros de tiempos tomados en situaciones simuladas.

## **CAPÍTULO 2: MARCO TEÓRICO**

### **2.1 BASES TEÓRICO CIENTÍFICAS**

“Timetabling, hace referencia a la calendarización y la organización específicamente de horarios de distintas organizaciones tales como, hospitales, universidades, etc., pero también es uno de los problemas más abarcados en la actualidad” (Guerra, Pardo y Salas, 2013, pág. 260) “(...) y un problema clásico del área de optimización combinatoria y es, así mismo, un problema de gran complejidad computacional. Ya que, pertenece a una familia de problemas conocidos como NP-Complejos, para los cuales no se conoce un algoritmo, que permita su resolución en tiempo polinomial” (Castañeda Ramírez, García Sánchez, Martínez Ruiz y Muñoz Arteaga, s.f.) y el tiempo de ejecución de este tipo de algoritmos crece a medida que aumentan los datos de entrada.

Timetabling consiste en asignar recursos a objetos ubicados en tiempo y espacio específicos teniendo en cuenta un conjunto de restricciones. Un ejemplo muy representativo es la generación de horarios de clase en una universidad, este proceso consiste en la asignación de recursos tales como profesores, aula y horarios a una clase específica, teniendo en cuenta que esta asignación debe cumplir ciertas condiciones para que sea considerada aceptable.

Debido a la necesidad de dar solución a este tipo de problemas y a la complejidad para desarrollarlos es que se ha recurrido a usar diferentes métodos heurísticos para darle solución, entre ellos podemos encontrar Recocido Simulado, Búsqueda Tabú, Colonia de Hormigas, Algoritmos genéticos, entre otros. Cada uno de los diferentes métodos aporta ciertas ventajas y desventajas en su utilización para la solución de dicho problema, “(...) se podría considerar a los algoritmos genéticos como buena opción para dar solución al problema de timetabling, (...) tiene varios puntos a favor comparado con otras heurísticas al momento de resolver problemas de timeabling.” (Naupari Quiroz y Rosales Gerónimo, 2010, pág. 116).

Como se mencionó, existen diferentes métodos tradicionales para resolver problemas de búsqueda y optimización, algunos métodos de optimización pueden desarrollar problemas de tamaño limitado dado que pueden degenerarse debido al creciente tiempo que necesitan para hallar la solución óptima, es por ello que para resolver este tipo de problemas

se hace uso de técnicas heurísticas. Los métodos heurísticos permiten explorar un espacio de soluciones e ir buscando soluciones que mejoren en cada iteración a la anterior. Cuando los resultados alcanzados por los métodos heurísticos no son los esperados, entonces es posible usar métodos metaheurísticos para obtener mejores resultados que los alcanzados por los heurísticos. Osman y Kelly (1995) citado por Riojas (2005) sostienen que:

Los procedimientos Metaheurísticos son una clase de métodos aproximados que están diseñados para resolver problemas difíciles de optimización combinatoria, en los que los heurísticos clásicos no son efectivos. Los Metaheurísticos proporcionan un marco general para crear nuevos algoritmos híbridos combinando diferentes conceptos derivados de la inteligencia artificial, la evolución biológica y los mecanismos estadísticos.

Los métodos metaheurísticos incluyen métodos evolutivos donde se encuentran los algoritmos genéticos (AGs), estos a su vez se encuentran dentro de la computación evolutiva en el campo de la Inteligencia artificial. “(...) y su fortaleza radica en que llevan a cabo una búsqueda paralela en el espacio de soluciones. (Forrest, 1993)” (Castañeda et al.) Los AGs se basan en la teoría de evolución de los seres vivos postulados por Darwin, Holland un profesor de la universidad de Michigan desarrolló a finales de los 60s una técnica llamada “planes reproductivos”, luego esta técnica se hizo conocida como AGs tras la publicación de su primer trabajo sobre el tema, llamado *Adaptation in Natural and Artificial Systems* en 1975.

Los AGs son métodos adaptativos usados para resolver problemas de búsqueda y optimización, Koza (1992) lo define como:

Es un algoritmo matemático altamente paralelo que transforma un conjunto de objetos matemáticos individuales con respecto al tiempo usando operaciones modeladas de acuerdo al principio Darwiniano de reproducción y supervivencia del más apto, y tras haberse presentado de forma natural una serie de operaciones genéticas de entre las que destaca la recombinación sexual. Cada uno de estos objetos matemáticos suele ser una cadena de caracteres (letras o números) de longitud fija que se ajusta al modelo de las cadenas de cromosomas, y se les asocia con una cierta función matemática que refleja su aptitud.

Los AGs se basan en el principio de supervivencia del más apto y el principio de selección natural, donde los individuos con mejores características tienen la posibilidad de

dejar descendencia y donde a lo largo de las generaciones las poblaciones evolucionan con la finalidad de sobrevivir en el entorno y donde las nuevas generaciones van adquiriendo ciertas características de los padres.

Una de las características principales de los AGs es explorar espacios de búsqueda en forma paralela y esto permite encontrar una solución en un tiempo competitivo a otras técnicas de optimización combinatoria. Como usan una analogía directa al principio postulado por Darwin es que se debe seguir una secuencia de pasos para poder aplicarlo a la resolución de problemas, entre estos tenemos:

- Generar una población inicial de manera aleatoria para garantizar la diversidad de la solución.
- Evaluar la aptitud de todos los individuos de la población tomando en cuenta alguna función objetivo (fitness).
- Crear una nueva población.
- Descartar la población vieja e iterar la nueva hasta que se dé una condición de término.

Para finalizar el AGs, es posible definir una condición de término, detenerlo cuando ya no se produzcan más cambios en la población, es decir cuando se dé la convergencia del algoritmo, fijar un número máximo de iteraciones antes de finalizar o cuando se halle la mejor solución. Antes de iniciar con la codificación el algoritmo es necesario definir ciertos parámetros:

- Tamaño de la población: Parámetro que indica la cantidad de individuos que tendrá la población para una generación determinada. Este parámetro debe ser lo suficientemente grande para garantizar la diversidad de los individuos, y a su vez pequeño para que el AGs no sea excesivamente lento. Arranz de La Peña y Parra Truyol (2014) mencionan que estudios revelan que hay un límite a partir del cual es ineficiente elevar el tamaño de la población puesto que no se consigue una mayor velocidad en la resolución del problema.
- Probabilidad de mutación: Parámetro que indica la frecuencia con la que los genes de un individuo pueden ser mutados, si la probabilidad es nula,

entonces esto indica que los individuos no sufrirán ningún cambio y serán idénticos a los padres. La probabilidad de mutación suele ser baja debido a que esto ocurre con poca frecuencia en la naturaleza.

- Probabilidad de cruce: Parámetro que indica la frecuencia con la que los individuos seleccionados compartirán su material genético.
- Criterio de parada: Especifica la condición que se tendrá en cuenta para terminar con la ejecución del algoritmo.

Fogel (1995) citado por Johnston (2000) menciona que la probabilidad de cruce normalmente se encuentra 0.6 a 0.95 mientras que el de mutación normalmente oscila entre 0.001 y 0.1.

La posible solución a un problema está compuesta por un conjunto de parámetros que definen las características de dichas soluciones. Alcaraz Soria, Maroto Álvarez, y Ruiz García (2002) mencionan que estos parámetros, conocido como genes, se pueden poner uno a continuación del otro formando una cadena de valores, a la que se hace referencia como cromosoma. Todos los operadores utilizados por un algoritmo genético se aplicarán sobre estos cromosomas y las operaciones que se realicen sobre estos cromosomas darán como resultado una nueva población, es por ello que determinar la codificación adecuada del cromosoma para la solución de un problema en específico es relevante en el desarrollo del AG. Luego de haber definido los parámetros del AG debemos elegir la codificación de las variables, aunque la codificación del AG simple propuesto por Holland usaba una codificación binaria se han ido desarrollando otros tipos de codificación, entre las mencionadas por Arranz de La Peña y Parra Truyol (2014) tenemos:

- Codificación Binaria: En este caso, cada cromosoma es una cadena de bits (0 o 1). Esta codificación permite abarcar muchos cromosomas incluso con un número reducido de genes. Sin embargo por otro lado esta opción no es la idónea para muchos problemas y en algunas ocasiones es necesario realizar correcciones después de la reproducción y/o mutación.
- Codificación Numérica: En este tipo de codificación se utilizan cadenas de números que representan un número en una secuencia. Se utiliza en problemas en los que hay que ordenar algo, donde resulta muy útil. En

algunos casos también es necesario como en el caso anterior realizar correcciones tras relaciones o mutaciones.

- **Codificación por Valor Directo:** Este tipo de codificación será el utilizado en caso de resolución de problemas en el que se requiera del uso de valores de cifrado complicado como podría ser en el uso de números reales, cuya codificación con números binarios sería muy complejo. En codificación por valor directo cada cromosoma es una cadena de valores relacionados con el problema a estudiar, pudiendo ser desde números decimales, cadenas de caracteres o incluso una combinación de varios de ellos.
- **Codificación en Árbol:** Este tipo de codificación se utiliza principalmente en el desarrollo de programas o expresiones para programación genética. Cada cromosoma será en este caso un árbol con ciertos objetos. En este método, los cambios aleatorios pueden generarse cambiando el operador, alterando el valor de un cierto nodo del árbol o simplemente sustituyendo un subárbol por otro.

Para poder realizar la reproducción de los individuos es necesario seleccionarlos teniendo en cuenta una probabilidad, los individuos seleccionados serán aquellos que muestren un mejor desempeño respecto a otros. “Por lo tanto una vez evaluado cada cromosoma y obtenida su puntuación, se tiene que crear la nueva población teniendo en cuenta que los buenos rasgos de los mejores se transmitan a ésta” (Arranz de La Peña y Parra Truyol, 2014). Aquellos individuos con un mejor desempeño serán aquellos que tendrán más posibilidad de transmitir su material genético. “Este operador no crea nuevas soluciones, simplemente tiene por objeto que pasen a la siguiente generación las buenas soluciones de la generación actual. (...). De este modo, (...) es responsable de la convergencia (...)” (Alfonso, Cazorla, Colomina, Escolano y Lozano. 2003, pág. 150). Los diferentes tipos de selección están divididos en tres grupos, aquellos basados en la calidad, en el orden y aquellos que realizan la selección mediante torneos. Entre los principales tipos de selección tenemos:

- **Selección por Rueda de Ruleta:** Método proporcional a la calidad cuya selección de padres depende de su adaptación es decir cada individuo tendrá una probabilidad proporcional a su calidad. Esta selección consiste en

colocar los cromosomas en la ruleta, ocupando estos un tamaño proporcional a la probabilidad que poseen, luego esta ruleta se hace girar y se selecciona el cromosoma en el que se para la ruleta. “El problema de este método es que cuando aparecen superindividuos (...), estos tienden a dominar rápidamente la misma y hacer converger al algoritmo prematuramente” (Alfonso et al., pág. 150).

- Selección estocástica universal: Es similar al método de la ruleta su diferencia radica en tener  $n$  marcadores igualmente distanciados y mediante una sola tirada se pueden seleccionar  $n$  individuos.
- Selección por Ranking: “Desarrollado por Whitley (1989) consiste en calcular las probabilidades de reproducción atendiendo a la ordenación de la población por el valor de adaptación en vez de atendiendo simplemente a su valor de adecuación” (Caballero Fernández, Gómez Nuñez, Luque Gallego, Molina Luque, y Torrico González, pág. 4). Este método puede evitar la convergencia prematura pero puede ser computacionalmente costosa debido a la necesidad de tener que ordenar la población.
- Selección Elitista: Consiste en copiar el cromosoma con mayor aptitud o alguno de los mejores a la nueva población con la finalidad de no perderlos aplicando cruce y mutación.
- Selección por torneo: En este tipo de selección se escogen aleatoriamente cierta cantidad de individuos (generalmente dos) y luego se determina quién gana el torneo. Para elegir al ganador de la competencia se tiene dos opciones, elegir al individuo mejor adaptado (selección por torneo determinista) o elegir al individuo de acuerdo a una probabilidad (selección por torneo probabilístico). Cuando el número de individuos del torneo es grande entonces se considera una selección dura teniendo los peores individuos escasas posibilidades de reproducirse.
- Selección sexual: Inspirado por el concepto de selección sexual propuesto por Darwin, dicha selección se hará separando entre hembras y machos. Este diseño asegura que todas las hembras se lleguen a reproducir independientemente de su aptitud y su objetivo es facilitar la exploración



del espacio de búsqueda. Por el otro lado, el sesgo hacia los machos explota el conocimiento de que la estructura del cromosoma del individuo es mejor (Arroyo Apaza, 2013, pág. 29).

- Selección Discriminatoria de individuos: Propuesto por Arroyo Apaza, se basa en el cruce de individuos que tengan un similar desempeño, para hacer uso de este operador es necesario dividir la población en clases de acuerdo a su calidad y además usa el Esquema de ponderación de Probabilidades de Mutación (EPPM).

Luego de haber seleccionado a los individuos se procede a la reproducción o crossover, esta operación consiste en intercambiar el material genético entre dos individuos y tiene como objetivo conseguir que el descendiente mejore la calidad de sus padres.

Para aplicar el cruce se deberá seleccionar con anterioridad dos individuos de la población con una de las diversas técnicas de selección. Además esta selección puede elegir el mismo padre para un descendiente, esto no es ningún problema pues se asegura la perpetuación del cromosoma más dominante pero si este cruce se produjese con mucha frecuencia podría acarrear consecuencias adversas en caso de que ese cromosoma dominante presente algunos genes no deseados. Para realizar el cruce se provee de diversas técnicas, entre las más usadas se encuentran:

- Cruce de un punto: Primero se procede a seleccionar la ubicación en las cadenas de los progenitores luego se intercambia los genes que se encuentran al lado izquierdo de esta posición.
- Cruce de n puntos: Se seleccionan n posiciones en las cadenas de los progenitores, luego se procede a intercambiarlos.
- Cruce uniforme: Donde cada gen del descendiente se elegirá de forma aleatoria entre cualquiera de los padres, para esto se hace uso de una máscara de 0 y 1, que especifican que padre aportara su información genética al primer y segundo descendiente.

Una forma de analizar los distintos métodos de selección es usando el concepto de presión selectiva. Las mejores soluciones ejercerán una presión selectiva en la población,

llenándola con más copias de sí mismo produciendo en muchos casos una convergencia prematura en la población.

Los métodos de selección se ordenan en orden creciente de presión selectiva (para valores estándares de sus parámetros) del siguiente modo: selección proporcional, ranking y torneo.

Después del cruce viene la mutación que “(...) consiste en cambiar el valor de una posición elegida aleatoriamente de un “string” (individuo)” (Universidad de Tarapacá ARICA-CHILE, 2000), es considerado un operador secundario debido a que se aplica con un menor porcentaje a la población y su efecto no es demasiado notable.

Las mutaciones permiten explorar áreas posiblemente no abordadas del espacio de búsqueda, permiten además salir del óptimo local cuando la población ha caído en convergencia. Por lo tanto la mutación consiste en modificar ciertos genes de forma aleatoria atendiendo a la probabilidad de mutación establecida con anterioridad. La mutación depende de la codificación y de la reproducción. Si se abusa de la mutación podemos caer en el uso del algoritmo genético como una simple búsqueda aleatoria. Existen diferentes métodos y entre ellos encontramos:

- Mutación de un bit: Solo un bit tiene la oportunidad de ser mutado y este bit es seleccionado de forma aleatoria.
- Mutación multibit: Donde todos los bit tienen la misma posibilidad de mutarse y la decisión es calculada independiente por cada uno de los bits.
- Mutación de un gen: Donde solo un gen tiene la posibilidad de ser mutado, si se decide la mutación la posición del gen a mutar es generado aleatoriamente.
- Mutación multigen: Igual que la mutación multibit, la diferencia radica en que en lugar de cambiar un solo bit se cambia un gen completo.
- Mutación de intercambio: Tipo de mutación que permite el intercambio de los valores de 2 genes o bits, dichos valores son escogidos aleatoriamente.

Existen otros operadores generadores de diversidad, entre ellos encontramos:

- Zap: Usar este operador consiste en cambiar un gen en vez de un solo bit.
- Creep: Aumenta o disminuye en 1 el valor de un gen.
- Transposición: Consiste en intercambiar los valores de 2 genes del mismo cromosoma sin afectar a los demás.

## **2.2 ANTECEDENTES DE LA INVESTIGACIÓN**

Naupari Quiroz & Rosales Gerónimo (2010), en su tesis desarrollada plantearon automatizar el proceso de generación de horarios de clase con un margen de error de 2% mediante el uso de algoritmos genéticos, utilizando el lenguaje de programación Java para la implementación de dicho algoritmo. El algoritmo construido para la automatización del proceso de generación de horarios de clase permitió encontrar soluciones aceptables dentro del margen de error definido por lo que se recomendó la implantación de este sistema en la Facultad de Ingeniería de Sistemas e Informática de la UNMSM.

En esta misma línea de investigación se encontró la tesis desarrollada por (Johnston Barrientos, 2000), cuya finalidad fue presentar un sistema computacional capaz de encontrar la asignación óptima de clases, maestros y horarios utilizando un algoritmo genético. Se realizaron pruebas al sistema tomando como datos de entrada los requerimientos en un período determinado en la carrera de Ingeniería en Sistemas Computacionales del Instituto Tecnológico de Nuevo Laredo y una lista de los maestros disponibles así como su horario. Con los resultados de estas pruebas se llegó a la conclusión que el algoritmo desarrollado encuentra soluciones con una mejor aptitud que la realizada de forma manual, además de reducir considerablemente el tiempo de elaboración de horarios y fácilmente puede ser adaptado a cualquier institución de educación superior.

Pacheco Agüero (2000), realizó un estudio de los métodos que existen para solucionar el problema de generación de horarios, luego del estudio realizado optó utilizar los algoritmos genéticos para dar solución a dicho problema aduciendo que lo utilizará porque combina rapidez y eficiencia, produciendo así buenos resultados y contribuyendo a la optimización de los recursos materiales y humanos con los que cuenta la Universidad Tecnológica de Mixteca.

## **CAPÍTULO 3: DISEÑO DEL ALGORITMO GENÉTICO**

### **3.1 ANÁLISIS DEL PROBLEMA**

En este capítulo se analiza el proceso de elaboración de la programación académica en la FII con el fin de identificar los datos involucrados, las variables de decisión, plantear la función objetivo y determinar qué condiciones se deben tener en cuenta para que la realización de este proceso culmine con los resultados esperados.

El proceso de elaboración de la programación académica empieza cuando el Secretario Académico envía a los Directores de Escuela un oficio solicitando la lista de los cursos que se abrirán en el siguiente semestre académico. Los Directores de Escuela envían una lista de dichos cursos determinando las necesidades de los alumnos por llevar tales cursos; el Secretario luego de haber recibido esta información debe determinar la manera de asignarles a cada uno de estos cursos un aula y grupo teniendo en consideración ciertas restricciones como que no debe existir solapamiento de estos cuando pertenecen a la misma escuela, un mismo semestre y a un mismo plan de estudios, además se debe considerar que un aula no puede tener asignado más de una clase en un mismo grupo, entre otras consideraciones. Luego de haber asignado un aula y horario a las clases que se dictarán en la FII estas son enviadas a los Jefes de los Departamentos correspondientes para que ellos se encarguen de determinar qué docentes dictarán dichos cursos.

La FII cuenta con 5 departamentos académicos las cuales reúnen a los docentes de disciplina afines, los docentes de estos departamentos dictan cursos no solo en la FII sino que también brindan servicio a otras facultades que lo requieran. Para la asignación de docentes a las clases, el Jefe de cada departamento realiza una reunión donde cada docente dependiendo de su preferencia por enseñar cierto curso y de su disponibilidad de horarios elige que cursos programados dictar, la carga mínima y máxima según la categoría del docente también son tomados en cuenta en esta etapa de asignación de docentes. Los cursos que no pertenecen a estos departamentos son enviados a los que corresponden, esperando que en dicho departamento se asigne un docente para dictar el curso. Luego la lista de docentes que enseñarán cada curso es enviada al Secretario Académico teniendo él que actualizar los datos.

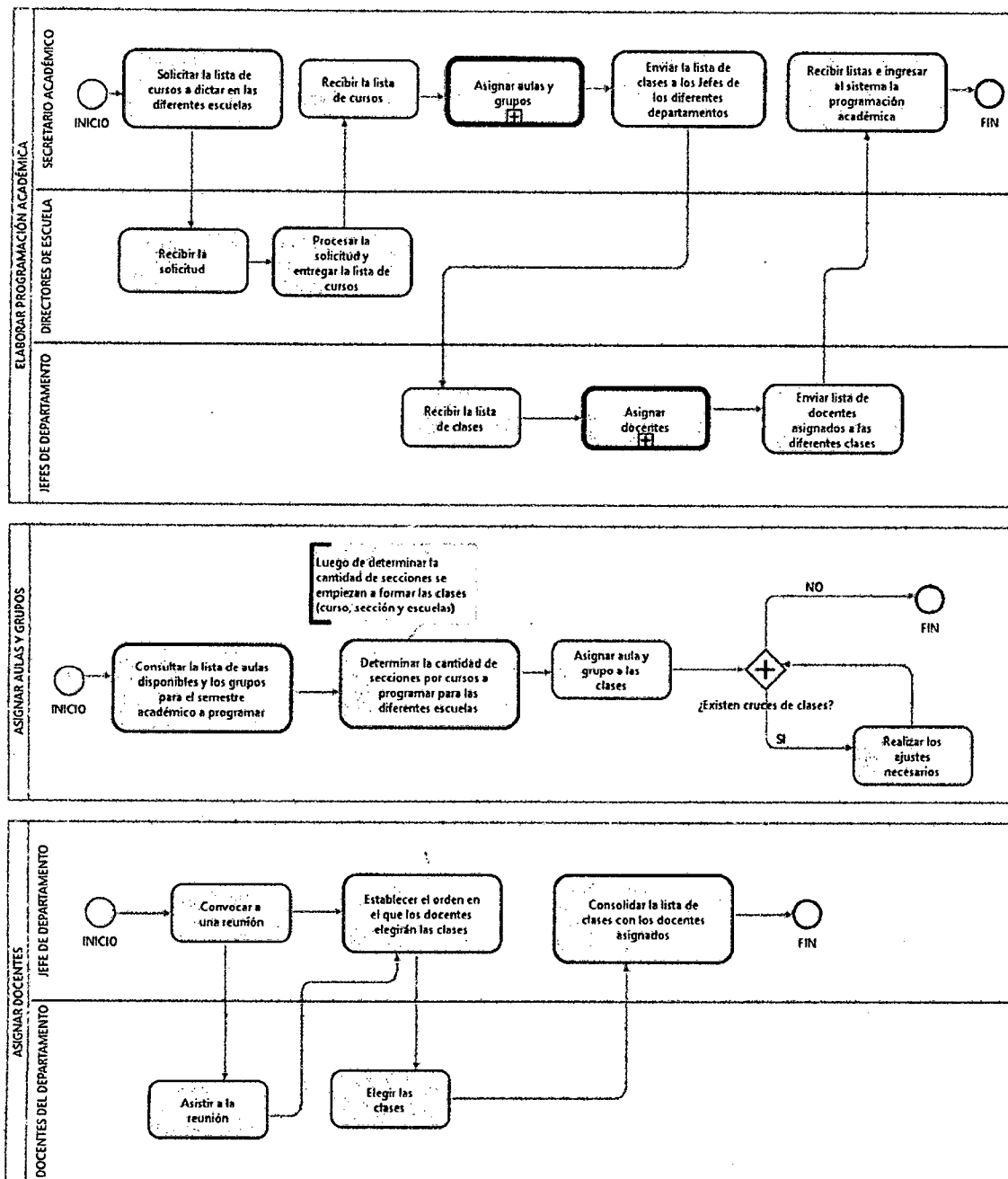


Figura 3.1 Diagrama de procesos para la elaboración de la P.A.

Fuente: Elaboración propia

### 3.1.1 Formulación del modelo matemático

Para poder definir el modelo matemático es necesario identificar los datos o elementos involucrados, variables, parámetros de decisión, restricciones y determinar la función objetivo correspondiente para la solución del presente problema.

- Conjunto de datos
  - Escuelas: Sea  $E = \{1, \dots, e\}$  el vector de escuelas pertenecientes a la FII.
  - Departamentos académicos: Sea  $DA = \{1, \dots, da\}$  el vector correspondiente a los diferentes departamentos académicos suscritos a la FII.
  - Docentes: Sea  $D = \{1, \dots, d\}$  el vector correspondiente a los docentes a los cuales se les puede asignar un curso.
  - Grupos: Sea  $G = \{1, \dots, g\}$  el vector de grupos en los cuales se podrá dictar los cursos.
  - Ciclos académicos: Sea  $CACD = \{1, \dots, cacd\}$  el vector correspondiente a los diferentes ciclos académicos que están asociados a los diferentes cursos que se dictan por escuela dependiendo del plan de estudios.
  - Plan de estudios: Sea  $PE = \{1, \dots, pe\}$  el vector correspondiente a los diferentes planes de estudios para las diferentes escuelas. Estos planes de estudios agruparán diferentes cursos que son dictados en diferentes ciclos académicos.
  - Cursos generales: Sea  $CG = \{1, \dots, cg\}$  el vector correspondiente a los cursos que son dictados por los diferentes departamentos académicos.
  - Cursos – Plan de Estudios: Sea CPE la matriz correspondiente a los cursos  $c$  que pertenecen al plan de estudio  $pe$ , cada uno de estos cursos del plan de estudios son dictados en un ciclo académico  $cacd$  específico.

- Cursos: Sea  $C = \{1, \dots, c\}$  el vector de cursos que se programarán en la FII, estos cursos son tomados de CPE teniendo cada uno de ellos información del plan para el que se dictará.
- Horas por curso: Sea  $HC = \{hc_1, \dots, hc_c\}$  el vector de horas correspondientes a cada uno de los cursos  $c$ .
- Secciones abiertas por curso: Sea  $SCPE$  el vector que indica la cantidad de secciones abiertas para un curso  $c$  que pertenece al plan de estudios  $pe$ .
- Tipos de aulas: Sea  $TA = \{1, \dots, ta\}$  el vector correspondiente a los diferentes tipos de aula en las que pueden ser dictados diferentes cursos, las aulas se clasifican en: normal (donde se dictaran dichos cursos que no requieren de ningún equipo adicional), laboratorio de computo, taller, laboratorio de mecatrónica, laboratorio de neumática.
- Cursos – Tipo de aula: Sea  $CTA$  la matriz correspondiente los cursos y tipos de aula que se quiere para el dictado de dichos cursos.
- Aulas: Sea  $A = \{1, \dots, a\}$  el vector de aulas que son usadas por la FII donde es factible dictar un curso, cada aula tiene un tipo  $TA$  dependiendo a sus características.
- Aula – Tipo: Sea  $AT$  el vector correspondiente al tipo de cada una de las aulas.
- Horarios de Docente: Sea  $HD$  la matriz correspondiente al horario disponible por los docentes de cada departamento.
- Docente por curso: Sea  $DC$  la matriz correspondiente a los docentes que enseñan el curso  $cg$ .
- Categoría: Sea  $CAT = \{1, \dots, cat\}$  el vector correspondiente a los diferentes tipos de categorías a los que pueden pertenecer los docentes.
- Dedicación: Sea  $DED = \{1, \dots, ded\}$  el vector correspondiente a los diferentes tipos de dedicación de los docentes.

- Horas por categoría: Sea  $HMIN = \{hmin_1, \dots, hmin_{md}\}$  el vector correspondiente a las horas mínimas de dictado de clase correspondiente a las diferentes categorías de los docentes.
- Parámetros: Aquellas variables que podemos controlar.

$p = \text{cantidad de aulas}$

$q = \text{cantidad de cursos,}$

$r = \text{cantidad de docentes,}$

$s = \text{cantidad de grupos}$

$$\forall_{p,q,r,s} \in N$$

Donde:

$p = \text{cantidad de aulas}$ , hace referencia a la cantidad de aulas disponibles para el dictado de clases en la FII.

$q = \text{cantidad de cursos}$ , hace referencia a los diferentes cursos a los cuales se les asignará un aula, grupo y docente. Estos cursos pueden pertenecer a los diferentes ciclos académicos y plan de estudios.

$r = \text{cantidad de docentes}$ , hace referencia a la cantidad de docentes suscritos a los diferentes departamentos académicos. La cantidad de docentes está determinada por aquellos a los cuales se les pueda asignar cursos.

$s = \text{cantidad de grupos}$ , hace referencia a la cantidad de grupos de horarios en los cuales se pueden programar cursos.

Nota: No se considera la capacidad de aulas debido a la frecuencia en la que dicha capacidad se modifica por la ampliación de cupos en las clases.



- Variables de decisión:

$X_{ijkl}$  = Representa asignar al aula  $i$  un curso  $j$  el cual es dictado por el docente  $k$  en el grupo  $l$ . Es una variable binaria (0,1).

$$i = 1, \dots, p$$

$$j = 1, \dots, q$$

$$k = 1, \dots, r$$

$$l = 1, \dots, s$$

$$X_{ijkl} \begin{cases} 0; & \text{si el aula } i \text{ no es asignada a un curso } j \\ & \text{dictado por el docente } k \text{ en el grupo } l \\ 1; & \text{en caso contrario} \end{cases}$$

- Restricciones:

Generalmente para problemas *timetabling* en la educación, se manejan 2 tipos de restricciones, Larrosa & Meseguer (2003) definen estos tipos de restricciones esencialmente diferentes como:

- Restricciones duras: Son condiciones de obligatorio cumplimiento tales como propiedades espaciales (p.ej. propiedades geométricas) o temporales (p.ej. un objeto no puede estar en dos lugares a la vez).
- Restricciones blandas: Son restricciones que en realidad denotan preferencias del usuario y se desea que se cumplan en la medida posible (p.ej. en un problema de generación de horarios no se desea que un profesor imparta dos clases consecutivas en aulas distantes).

Para el caso de la FII-UNP se contemplarán las siguientes restricciones:

**Restricciones duras u obligatorias (RO):** En caso de ser violadas generan un horario inválido.

- No se debe asignar un docente a un curso que no sea de su preferencia (RO1).

$$X_{ijkl} = DC_{kj} \quad \forall X_{ijpq} = 1, \forall i \in A, \forall j \in C, \forall k \in D, \forall l \in G$$

- Un docente debe impartir una cantidad determinada de horas de dictado de clase como mínimo según sea su categoría (RO2).

$$\sum_{i=1}^p \sum_{k=1}^r \sum_{j=1}^s X_{ijkl} * HC_j = HMIN_k$$

Nota: Esta restricción se cumple dependiendo de la disponibilidad de cursos por asignar al docente k. La asignación de carga académica se realiza en un orden establecido por la categoría del docente y los años de servicio.

- No se debe asignar un curso a un docente fuera de su disponibilidad de horarios (RO3).

$$X_{ijkl} = HD_{kl} \quad \forall X_{ijkl} = 1, \forall i \in A, \forall j \in C, \forall k \in D, \forall l \in G$$

- Un docente puede tener asignado como máximo un curso en mismo grupo (RO4).

$$\sum_{k=1}^r \sum_{l=1}^s X_{ijkl} \leq 1 \quad \forall i \in A, \forall j \in C, \forall k \in D, \forall l \in G$$

- No deben coincidir los horarios de los cursos que corresponden a un mismo ciclo académico (CACD) y plan de estudios (PE) (RO5).

Los cursos que pertenecen al plan de estudios PE y son dictados en el ciclo CACD se encuentran en un subconjunto de la matriz CPE, por tanto se puede decir que existe un rango  $[a, b]$  de cursos que están incluidos en CPE.

#### **Para los cursos de tipo obligatorio**

Existe un rango de cursos  $[c, d] \subset [a, b]$

Considerando la programación de más de una sección para un mismo curso es que surgen diferentes combinaciones de grupos en los que el alumno puede llevar sus clases. El número de combinaciones está dado por:

$$Total\ de\ combinaciones = \prod_{scpe=c}^d SCPE_{scpe}$$

Existen diferentes combinaciones (COM) pero solo se evaluarán aquellas cuyo horario no coincidan, por tanto aquellas combinaciones deben cumplir lo siguiente:

$$\sum_{l=1}^s X_{ijkl} \leq 1; \forall X_{ijkl} \in COMB, \forall i \in A, \forall j \in C, \forall k \in D, \forall l \in G$$

Las combinaciones de clase que cumplen este requisito se encuentran en COMB, luego es necesario calcular lo siguiente:

$$\sum_{j=c}^d \sum_{l=1}^s X_{ijkl}, cada\ información\ es\ guardada\ en\ SUM_j$$

Para evaluar la restricción se debe cumplir:

$$\sum_{j=c}^d \sum_{l=1}^s X_{ijkl} = SUM_j$$

$$\forall [c, d] \subset [a, b]$$

$$\forall [a, b] \subset CPE \wedge \forall [a, b] \subset C, \forall pe \in PE$$

$$\forall cacd \in CACD, \forall i \in A, \forall j \in C, \forall k \in D, \forall l \in G$$

**Para los cursos de tipo electivo**

Existe un rango de cursos  $[e, f] \subset [a, b]$

Para el caso de los cursos de tipo electivo se determina si dicho curso puede ser llevado en alguna de las combinaciones disponibles.

- La cantidad de cursos que requieran cierto tipo de aula no debe sobrepasar la cantidad de aulas del mismo tipo (RO6).

$$\sum_{l=1}^s X_{ijkl} \leq \sum A_{ta} ; CTA_j = ta$$

$$\forall i \in A, \forall j \in C, \forall k \in D, \forall l \in G, \forall ta \in TA$$

- No se debe asignar más de un curso a un aula en un grupo específico (RO7).

$$\sum_{l=1}^s \sum_{i=1}^p X_{ijkl} \leq 1 \quad \forall i \in A, \forall j \in C, \forall k \in D, \forall l \in G$$

- Un aula puede tener como máximo un curso y un docente asignados en un grupo específico (RO8).

$$\sum_{j=1}^q \sum_{k=1}^r X_{ijkl} \leq 1 \quad \forall i \in A, \forall j \in C, \forall k \in D, \forall l \in G$$

- Los cursos de cierto tipo (laboratorio, taller, etc.) deben ser asignados a aulas correspondientes a su mismo tipo (RO9).

$$CTA_j = AT_i \quad \forall X_{ijkl} = 1, \forall i \in A, \forall j \in C, \forall k \in D, \forall l \in G$$

- La cantidad de grupos de un curso debe ser igual a la cantidad de secciones que se desea programar de dicho curso (RO10).

$$\sum_{l=1}^s \sum_{j=1}^q X_{ijkl} = SCPE_{jpe}$$

$$\forall i \in A, \forall j \in C, \forall k \in D, \forall l \in G$$

**Restricciones blandas o deseables:** La violación de algunas de ellas seguirá produciendo un horario válido, pero de menor calidad.

- Los cursos son programados en los turnos mañana, tarde y noche según el ciclo al que pertenecen (los cursos que pertenecen al ciclo IX hacia adelante son programados de preferencia en turno noche).

$$x_{ijkl} = TC_l; \quad x_{ijkl} = 1, \forall i \in A, \forall j \in C, \forall k \in D, \forall l \in G$$

Nota: Esta restricción no es tomada en cuenta para su evaluación porque considerarla implica la omisión de algunos valores en las variables, por tanto el espacio de soluciones se reduciría y en algunas situaciones puede no encontrarse una solución factible.

- Función objetivo:

Para poder asegurar que las soluciones encontradas son factibles, es preciso usar alguna función que lo determine (función objetivo), en este caso para dar solución al problema de tipo *timetabling* el objetivo es minimizar el cruce que existe entre las aulas y docentes en mismo grupo. Por lo tanto el objetivo del modelo matemático descrito será encontrar una solución óptima donde el incumplimiento de las restricciones sea la mínima posible o en el mejor caso nula. La función objetivo para este problema sería:

$$\min f(x) = cfa(x) + cfd(x) + cfg(x) \quad \forall x \in \{1, \dots, N\}$$

Donde:

*X*; representa a cada elemento de la matriz de soluciones

*N*; representa al N° de elementos en la matriz de 4 dimensiones

*cfa(x)*; representa los conflictos de aulas

*cfd(x)*; representa los conflictos de docentes

*cfg(x)*; representa los conflictos de grupo

### 3.2 DISEÑO DEL CROMOSOMA

Una etapa importante en la elaboración del AGs es la codificación de la información de cada solución en una cadena llamada cromosoma, encontrar la representación adecuada

del cromosoma influye en los resultados. Los AGs trabajan sobre una población que está formada por individuos (posibles soluciones al problema), estos a su vez están formados por genes, en término biológicos al conjunto de genes se les conoce como genotipos y son los genotipos quienes contienen la información requerida para formar un nuevo individuo.

Comparando estos términos se logró identificar la representación al problema, la programación académica viene a ser el individuo (solución al problema), dentro de la programación académica se encuentran las clases (unidad de asignación), a estas clases se les asigna recursos tales como docente, aula y grupo, el conjunto de recursos asignados a cada clase vienen a ser los genes y dichos genes forman el cromosoma. Las clases son objetos cuyas propiedades nos brindan información sobre el curso, la(s) escuela(s) para la(s) que es programado el curso, el tipo de aula que necesita dicho curso para ser dictado y al finalizar el algoritmo también brinda información sobre el docente, grupo, sección y aula asignada.

Como se mencionó, la unidad de asignación son las clases y la cantidad de estas clases es determinada por la cantidad de cursos que necesiten programarse en los diferentes semestres académicos.

$Clase_1$	$Clase_2$	$Clase_3$	...	$Clase_{N-2}$	$Clase_{N-1}$	$Clase_N$
-----------	-----------	-----------	-----	---------------	---------------	-----------

**Figura 3.2 Unidad de asignación**

*Fuente: Elaboración propia*

Los recursos que necesitan asignarse se muestran en la Figura 3.2 y para determinar qué procedimiento se usaría en dichas asignaciones fue necesario seguir el proceso realizado manualmente. Sabiendo que la asignación de docentes se realiza en cada departamento académico y además está sujeta a procedimientos que no pueden ser simulados con los AGs es que se determinó que dicha asignación debería seguir un procedimiento secuencial y que solo la asignación de grupos y aulas deberían usar AGs.

$Docente_{clase1}$	$Docente_{clase2}$	...	$Docente_{claseN-1}$	$Docente_{claseN}$
$Grupo_{clase1}$	$Grupo_{clase2}$	...	$Grupo_{claseN-1}$	$Grupo_{claseN}$
$Aula_{clase1}$	$Aula_{clase2}$	...	$Aula_{claseN-1}$	$Aula_{claseN}$

**Figura 3.3 Recursos asignables**

*Fuente: Elaboración propia*

Cada gen del cromosoma contiene información de los 2 recursos que son necesarios asignar, al principio se intentó asignar los 3 recursos para cada clase, la solución intentó encontrar el mejor individuo pero no todas las restricciones se llegaron a cumplir, debido a que a veces solo era necesario cambiar un solo recurso para que las soluciones mejoren su calidad, luego de haber hecho estas pruebas se optó por seguir la recomendación de Naupari Quiroz et al. (2010) quienes hacen uso de una variación del método “divide y vencerás” con la finalidad de mejorar la heurística.

Son 3 los recursos que se deben asignar y cada uno de estos recursos se asignaron en fases y procedimientos diferentes, se considera la disponibilidad de horarios de los docentes, es por ello que para poder evaluar las restricciones referentes a los horarios es necesario tener la información de los docentes, por lo que se deduce que la etapa de asignación de docentes es previa a la de asignación de horarios. Con la asignación de aulas es necesario considerar algunas restricciones como la que hace referencia a no asignar la misma aula a un grupo específico y con ello podemos deducir que para evaluar esta restricción es necesario tener la información de los grupos, es decir la etapa de asignación de aulas sería la última.

Existen diferentes tipos de codificación y cada una de ellas es seleccionada dependiendo de la información que desea representar. Para dar solución al problema es necesario tener una referencia a los recursos que serán asignados porque no sería una buena opción trabajar con los recursos directamente (codificación por valor directo) debido a que se tendría que trabajar con objetos y el espacio ocupado por ellos es mayor, por lo tanto se optó por usar una codificación entera teniendo cada uno de los genes una referencia al recurso que será asignado. El valor de la referencia al recurso está dado por la ubicación de dicho recurso en la lista donde se encuentran todos los recursos dependiendo de la etapa de asignación en la que se encuentre.

<i>Ref.Recurso<sub>clase1</sub></i>	...	<i>Ref.Recurso<sub>claseN</sub></i>
-------------------------------------	-----	-------------------------------------

**Figura 3.4 Diseño del cromosoma**

*Fuente: Elaboración propia*

### 3.3 IMPLEMENTACIÓN DE LA FUNCIÓN DE APTITUD

La función de aptitud es la función objetivo en los problemas de optimización; dicha función debe reflejar los aspectos más relevantes del problema; estableciéndose las

condiciones que restringen los resultados proporcionados por el algoritmo. El valor de la función de aptitud representa la calidad de la solución o el *fitness* de cada individuo y dependiendo del valor que devuelva esta función se establecerá una probabilidad con la que el individuo tendrá la posibilidad de reproducirse, mientras mayor sea su adaptación más posibilidades de generar descendencia tendrá.

El objetivo fue encontrar aquella solución que cumpla con las restricciones establecidas, cuando la aptitud del individuo es 1 esto quiere decir que dicho individuo no incumple ninguna restricción y por tanto es la mejor solución. Otras soluciones aceptables son aquellas cuya aptitud esté más cerca de uno. En cada etapa se evaluó aquellas restricciones referentes al recurso que se asignaba, cada vez que el individuo incumple una restricción es penalizado, esta penalización depende de la cantidad de veces que el AG realiza una asignación no adecuada del recurso a cada una de las clases, esto se puede representar de la siguiente forma:

$$Ap_{ri} = \frac{\text{total de clases que incumplen } ri}{\text{total de clases que se espera que no incumplan } ri}$$

Donde:

*Ap<sub>ri</sub>*: Aptitud del individuo referente al cumplimiento de la restricción "i"

*ri*: Hace referencia a la restricción "i" que está siendo evaluada.

Cada restricción tiene una penalización dependiendo del tipo de restricción que está siendo evaluada, ya sea una restricción obligatoria o deseable, para este caso se consideró solamente las restricciones obligatorias por lo que no fue necesario asignar un peso de penalización a cada restricción y por tanto la aptitud final del individuo quedó expresada por:

$$f_{ev} = 1 - \frac{\sum_{i=1}^n Ap_{ri}}{n}$$

Donde:

*f<sub>ev</sub>*: Función de evaluación de un individuo

*n*: Número de restricciones evaluadas



*Ap<sub>ri</sub>: Aptitud del individuo referente al cumplimiento de la restricción "i"*

### **3.4 SELECCIÓN DE OPERADORES**

#### **3.4.1 Selección**

La selección permite determinar qué individuos son los que sobrevivirán y podrán generar descendencia de acuerdo a su capacidad de adaptación. Para escoger el método de selección que mejor se adapte a la solución del problema fue necesario considerar la presión selectiva de dichos métodos, considerando de esta manera a la selección por torneo como un método a usar. También se consideró utilizar la Selección discriminatoria de individuos bajo un esquema de ponderación de probabilidades de Mutación, modelo planteado por Arroyo Apaza (2013). Además de estos métodos se consideró probar con el método tradicional de ruleta.

Los 3 métodos fueron sometidos a diferentes pruebas y para la elección del método se consideró la presión selectiva, el promedio de generaciones necesarias para que dicho método logre encontrar la mejor solución y la frecuencia de éxito, siendo el método de selección por Torneo aquel que logró imponerse.

#### **3.4.2 Cruce**

Este operador principal trabaja sobre los individuos previamente seleccionados y su trabajo consiste en intercambiar información de los padres recombinando sus genes.

Para la elección del método se consideró el promedio de generaciones para encontrar la mejor solución. Se evaluó dos métodos, cruce por dos puntos y cruce uniforme, ambos métodos presentaron un buen performance y la diferencia entre ellos radica en el tiempo usado para encontrar dicha solución, los dos métodos mostraron un comportamiento variado en las diferentes etapas y una convergencia prematura para poblaciones pequeñas es por ello que se optó por realizar pruebas combinando dichos métodos, encontrando que la combinación de ellos producían mejores resultados. Usar cruce por dos puntos resultó más beneficioso usándolo en la etapa de asignación de grupos, mientras que cruce uniforme se adecuó mejor a la etapa de asignación de aulas.

### **3.4.3 Mutación**

Los cromosomas en cada una de las fases son formados aleatoriamente y sus genes adquieren valores de acuerdo a cierto rango dependiendo de las características de la clase a la que se asigna cada recurso. Considerando la formación del cromosoma se descarta el uso de la mutación por intercambio debido a que algunas veces no es posible intercambiar los recursos porque las características de las clases a los que serán asignados son diferentes.

Para la elección del método de mutación se analizó la mutación multigen y la mutación de un gen, resultando más beneficiosa la mutación de un gen. El proceso que se sigue para realizar la mutación del gen considera la clase a la que se le va a asignar un recurso para luego realizar la asignación de acuerdo a sus características.

### **3.4.4 Zap**

Cuando se detecta que la población tiende a tener una misma aptitud sin encontrar un individuo con solución óptima se dice que la población está estancada en un óptimo local, es por ello que para evitar el estancamiento se hace uso del operador zap, que permite sacar a la población de estancamiento, la variante de este operador es que incluye nuevos individuos a la población con la finalidad de explorar nuevos espacios de búsqueda y encontrar una solución óptima.

## **3.5 DESARROLLO DEL PROTOTIPO**

Una vez establecido el formato del cromosoma, de haber determinado la función de evaluación y los métodos para los diferentes operadores es necesario determinar qué datos son necesarios para la implementación del prototipo propuesto.

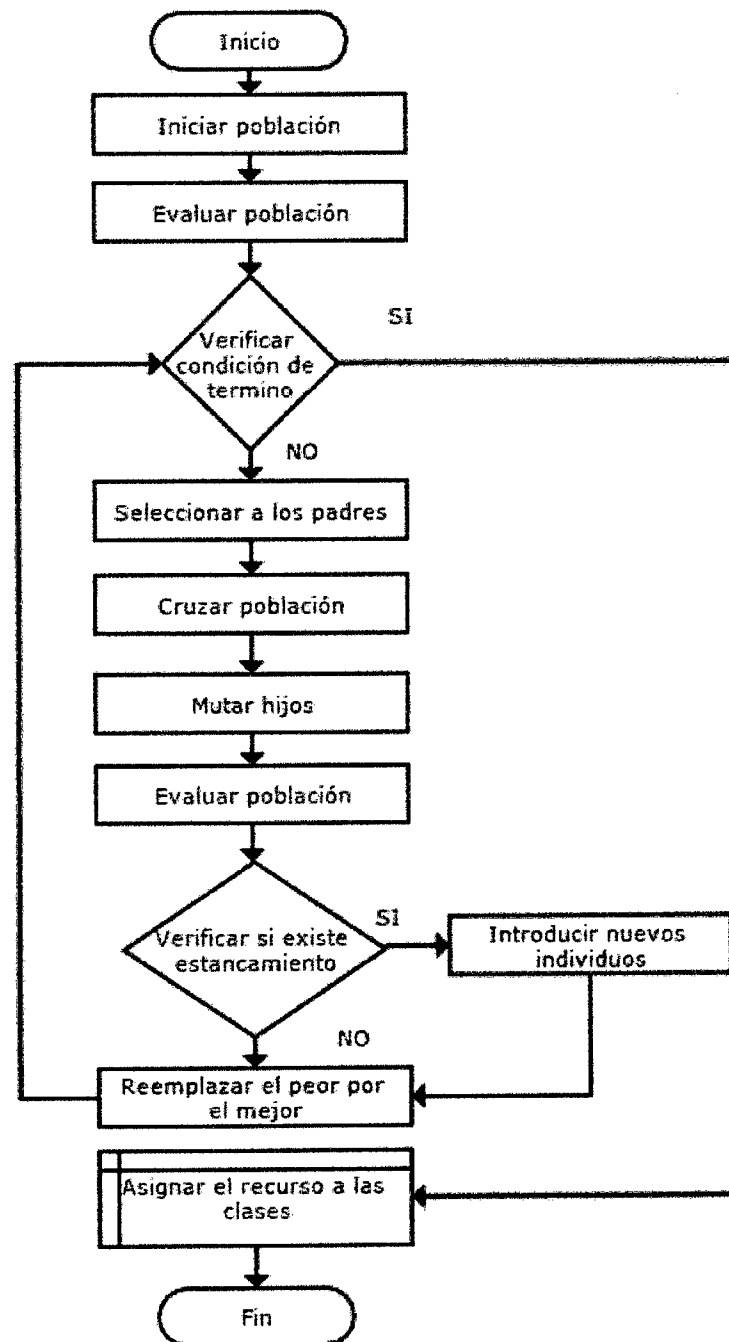
### **3.5.1 Determinación de los parámetros de entrada**

- Datos de entrada referentes al problema

Estos datos están relacionados a la configuración de las aulas, grupos de horario, los datos referentes a los docentes, la cantidad de clases y el tipo de aula que necesitan dichas clases para su dictado.

- Datos de entrada relativos al funcionamiento del algoritmo
  - Tamaño de población: Número entero múltiplo de dos, permite determinar la cantidad de soluciones por generación, su valor puede variar entre 500 y 800.
  - Número de generaciones: Número entero que determina la cantidad de iteraciones del algoritmo, su valor varía entre 500 y 20000.
  - Porcentaje de cruce: Su valor recomendado oscila entre 85% y 95%.
  - Porcentaje de mutación: Su valor se encuentre en el intervalo de 3% y 6%.

### 3.5.2 Diagrama de flujo del AGs.



*Figura 3.5 Diagrama de flujo del AGs*

*Fuente: Elaboración propia*

### 3.6 IMPLEMENTACIÓN DEL PROTOTIPO

Para explicar a detalle la implementación del prototipo se presenta un ejemplo donde se programarán 10 clases, se parte que los datos necesarios han sido previamente configurados o ingresados.

Categorías de docentes		
Nº	Categoría	Horas Mínimas
1	Principal	10
2	Auxiliar	15
3	Asociado	17

Aulas			
Nº	Posición	Aula	Tipo
1	0	I-1	Normal
2	1	I-2	Normal
3	2	Lab. HP	Lab. Cómputo

Grupos		
Nº	Posición	Grupo
1	0	01
2	1	02
3	2	03
4	3	04

Docentes						
Nº	Posición	Docente	Categoría	Antigüedad	Grupos	Cursos
1	0	DD0001	Principal	25	02,03	IO3412,IO3420
2	1	QR0002	Auxiliar	15	02,03,04	IO3420,IO4405
3	2	RG0003	Asociado	25	03,04	SI5492,SI5418
4	3	CA0004	Auxiliar	10		SI5441,SI5448

Nº	Código	Horas	Tipo	Plan	Ciclo	Aula
1	IO3420 - Modelos Estocásticos	5	O	2010-Industrial	V	Normal
2	IO3420	5	O	2010-Industrial	V	Normal
3	IO3412 - Investigación de Operaciones I	5	O	2010-Industrial	V	Normal
4	IO3412	5	O	2010-Industrial	V	Normal
5	SI3332 - Programación Multimedia	5	E	2010-Informática	VI	Lab. Cómputo
6	SI3332 - Programación Multimedia	5	E	2010-Informática	VI	Lab. Cómputo
7	SI3418 - Sistemas Operativos	5	O	2010-Informática	VI	Normal
8	SI3418 - Sistemas Operativos	5	O	2010-Informática	VI	Normal
9	SI5492 - Proyecto de tesis	8	O	2010-Informática	IX	Normal
10	SI5441 - Sistema de control y auditoria Inf.	5	O	2010-Informática	IX	Normal

### 3.6.1 Asignación de docentes

La asignación de docentes en la FII se da en cada uno de los departamentos y son los docentes quienes eligen sus clases dependiendo de los cursos que enseñan en base a su experiencia, cada docente va eligiendo dichas clases de acuerdo a un orden establecido teniendo en cuenta la categoría y años de antigüedad, es por ello que la asignación de docentes en este problema se realiza directa y secuencialmente.

Para realizar la asignación, los docentes deben estar ordenados por categoría y años de antigüedad en forma decreciente, siendo el docente de cierta categoría con más años de docencia quien se encuentre primero para la asignación. Para determinar qué clases y cuántas de ellas es necesario tener en cuenta las horas mínimas dependiendo de la categoría del docente.

El algoritmo va asignando a cada docente las clases y para ello hace uso del procedimiento “getMejorAsig” que devuelve las posiciones de las clases que se asignarán a

un docente específico. Algunas veces esta función no devolverá ninguna posición indicando que no existen clases disponibles para asignar a dicho docente. Una vez estando en el procedimiento “getMejorAsig” se llama al procedimiento “getPosiblesCursos” que devuelve la posición de las posibles clases que el docente puede impartir, cuando esta función devuelve posiciones de clases entonces se procederá a agrupar dichas clases de acuerdo a la cantidad de horas necesarias para su dictado. Luego de esto se determina la mejor secuencia de horas cuya suma sea igual o se aproxime a las horas mínimas del docente, como las clases se agruparon de acuerdo a las horas que necesitan, se escoge aleatoriamente una de las clases dentro del grupo de acuerdo a la hora especificada en la secuencia.

Como la elección de clases de acuerdo a las horas se hace de manera aleatoria se consideró la necesidad de realizar este proceso varias veces con la finalidad de encontrar la asignación que tenga más clases con docentes asignados. Al finalizar esta etapa las clases tendrán asignadas un docente y tal asignación cumple con las restricciones RO1 y RO2.

La asignación de docentes para el ejemplo seguido se muestra en la Tabla 3.1.

**Tabla 3.1**

*Asignación de docentes*

<b>Clase</b>	<b>Pos. -Docente</b>	<b>Docente</b>
Nº1	1	QR0002
Nº2	1	QR0002
Nº3	0	DD0001
Nº4	0	DD0001
Nº5	X	-
Nº6	X	-
Nº7	2	CA0004
Nº8	X	-
Nº9	2	CA0004
Nº10	3	CA0004

### 3.6.2 Asignación de grupos

En esta etapa se asigna los grupos de horario a las diferentes clases y cada asignación de la clase considera la lista de horarios disponibles del docente que ha sido asignado en la etapa anterior. En esta asignación las soluciones deben considerar

obligatoriamente el cumplimiento de las restricciones RO3, RO4, RO5 y RO6; y el procedimiento seguido se detalla a continuación.

**Iniciar población:** En problemas basados en restricciones es necesario delimitar las nuevas soluciones generadas, pues no todas son válidas, es por ello que para asignar los grupos a las clases se tiene en cuenta la asignación de docentes.

La población de soluciones es generada al azar y el valor de sus genes corresponde a cualquier posición generada aleatoriamente dentro del rango de la lista “grupos”. Algunos valores de los genes están delimitados dependiendo del docente que ha sido asignado generando así una población que cumpla con la restricción RO3 siendo innecesaria su posterior evaluación.

La implementación de esta etapa fue codificada en el procedimiento “iniciarPoblacionG” y para la creación de las posibles soluciones se tuvo en cuenta los resultados de la etapa anterior. Para dar valor a cada gen del cromosoma es necesario obtener el valor de la asignación docente y cuando este indique que no existe un docente asignado, entonces se obtiene un valor aleatorio dentro de la lista “grupos”, caso contrario se genera una posición aleatoria dentro de la lista “posiciones” llenada por el procedimiento “existenGruposDoc” que recibe como parámetro el código del docente. La lista “posiciones” es llenada con las posiciones reales de los grupos que el docente asignado tiene disponible, como ejemplo se presenta la Figura 3.5 donde se puede observar el procedimiento para la asignación de los valores a cada uno de los genes del primer individuo.



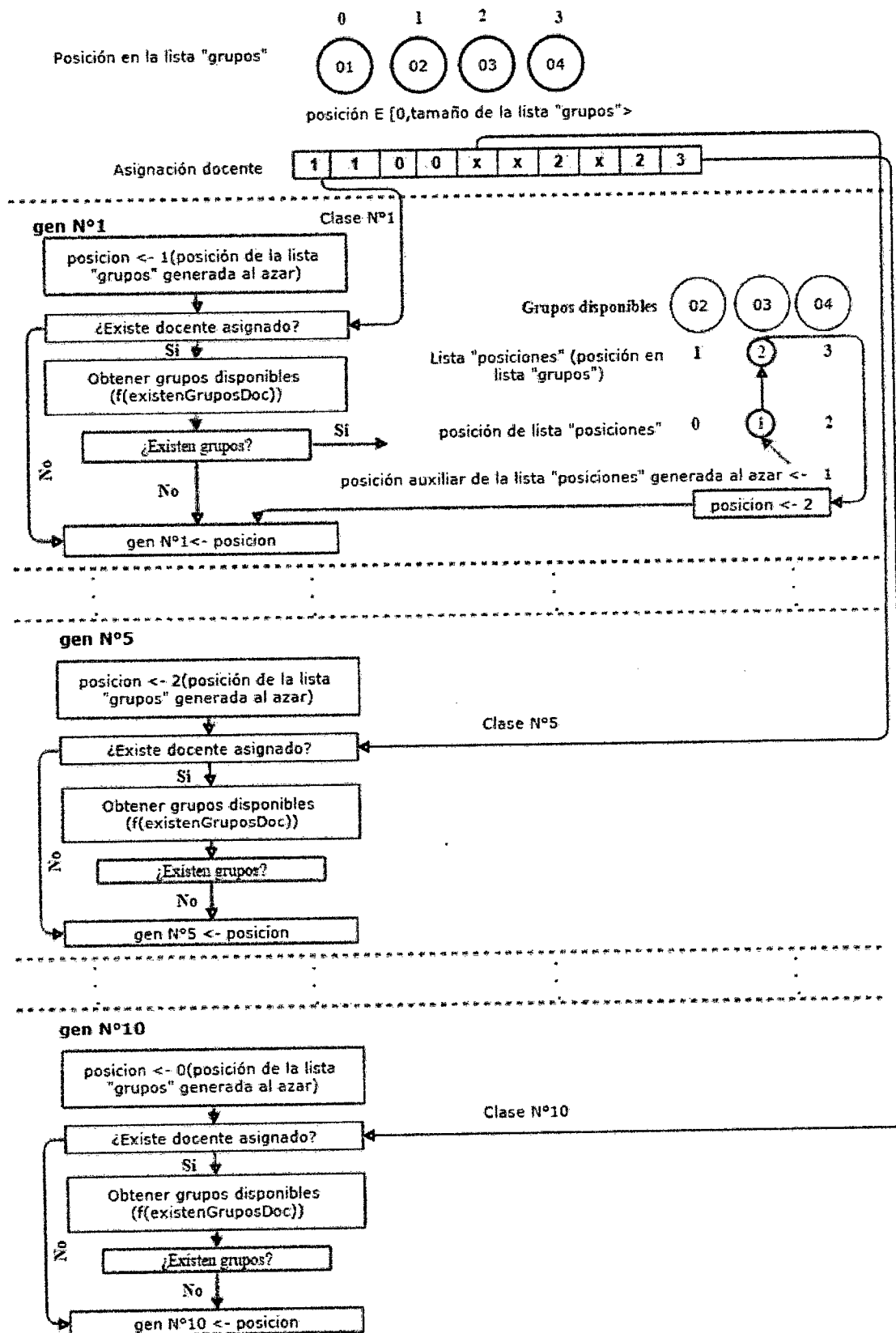


Figura 3.6 Pasos para generar un individuo – Asignación de grupos

Fuente: Elaboración propia

El procedimiento observado en la imagen es realizado para formar cada uno de los individuos de la población y para seguir con el ejemplo se generó los individuos mostrados en la Figura 3.6.

Indv N°1	2	3	2	1	2	2	2	0	2	0
Indv N°2	3	1	1	2	2	1	2	2	3	1
Indv N°3	1	3	1	2	2	3	2	0	3	1
Indv N°4	2	1	1	1	2	2	3	3	3	3
Indv N°5	1	3	2	1	1	1	3	3	3	3
Indv N°6	1	1	2	1	2	0	2	3	2	2

Figura 3.7 Población inicial - Asignación de grupos

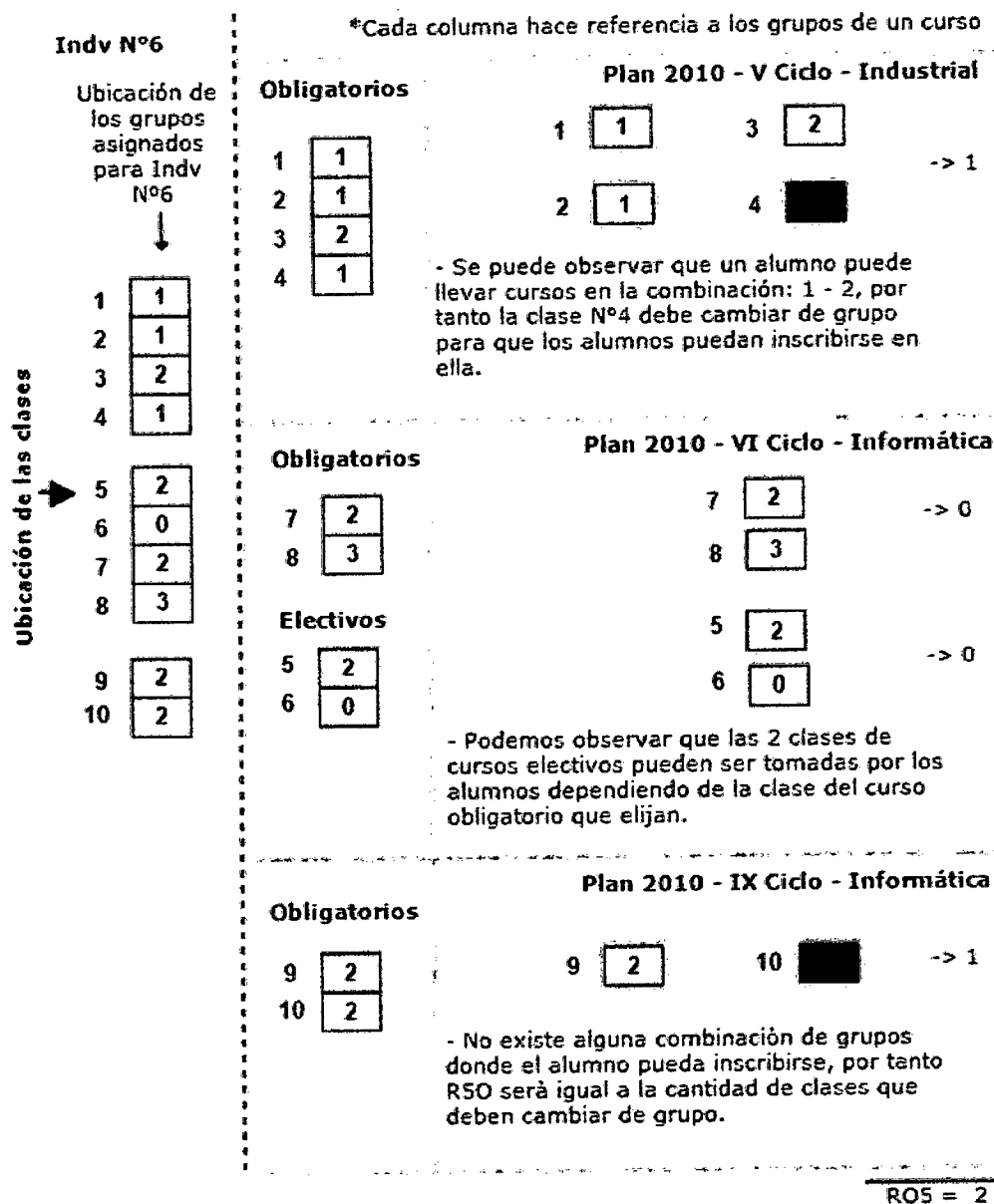
Fuente: Elaboración propia

**Evaluar la población:** Esta tarea fue implementada en el método “evaluarPobl”, devolviendo “verdadero” cuando detecta que la calidad del 97% de la población tiende a ser la misma con la finalidad de introducir nuevos elementos y evitar el estancamiento de las soluciones. Cada vez que se evalúa la población se invoca al método “evaluarR4R5R6xInd” siendo este el encargado de evaluar las restricciones RO4, RO5 y RO6 en cada uno de los individuos. En la Figura 3.7, Figura 3.8 y Figura 3.9 se muestra un esquema donde se evalúan las restricciones correspondientes para el último individuo.

Indv N°6		RO4:	
Ubicación de los docentes asignados		*Verificamos que las clases que son dictados por el mismo docente no tengan el mismo grupo para ser dictados.	
Ubicación de las clases		Ubicación de los grupos asignados para Indv N°6	
1	1	1	1
2	1	1	
3	0	2	2
4	0	1	1
5	x	2	
6	x	0	2
7	2	2	
8	x	3	2
9	2	2	
10	3	2	2

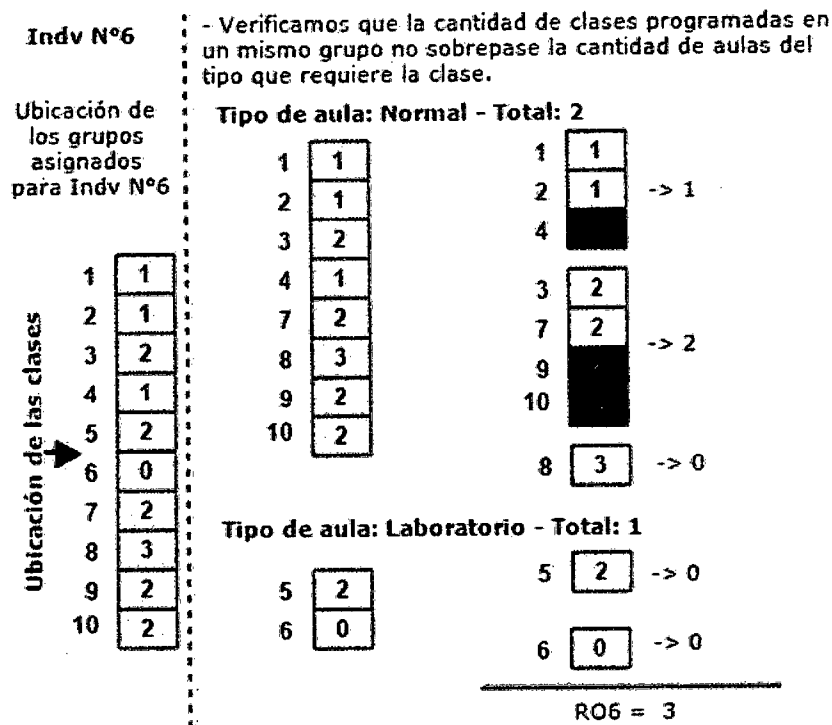
Figura 3.8 Evaluación de RO4

Fuente: Elaboración propia



**Figura 3.9 Evaluación de R05**

*Fuente: Elaboración propia*



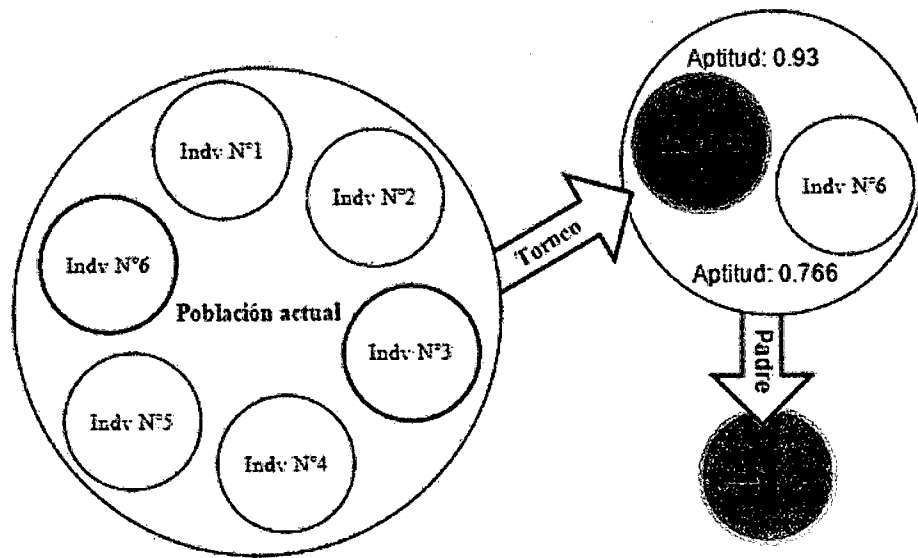
**Figura 3.10 Evaluación de RO6**

*Fuente: Elaboración propia*

La aptitud de los individuos para la primera población fueron 0.87, 0.87, 0.93, 0.77, 0.8 y 0.76 para el primer, segundo, tercero, cuarto, quinto y sexto individuo.

**Verificar condición de término:** En esta etapa del algoritmo se verifica si existe una solución óptima con aptitud 1 o si el número de generaciones del algoritmo se han cumplido. Para verificar la existencia de una buena solución se implementó la función “*existeBuenaSol()*” que devuelve “verdadero” cuando esta solución es encontrada y además guarda el individuo con mejor adaptación para luego aplicar elitismo.

**Seleccionar padres:** Hace uso del procedimiento “*seleccionarPadres*”, este método implementa la selección por torneo usando 2 individuos, el mejor de ellos es seleccionado para ser padre, el procedimiento seguido lo podemos observar en la Figura 3.11, este proceso es realizado tantas veces hasta llenar la población de padres.



**Figura 3.11 Selección por torneo**

*Fuente: Elaboración propia*

**Cruzar los padres:** Se hace uso de los padres seleccionados anteriormente, este proceso es realizado por el método “cruzarPoblacion” y tiene en cuenta el porcentaje de cruce para poder determinar si los padres intercambiaran su material genético o pasan tal cual a la siguiente generación.

Para el ejemplo consideramos que los 2 primeros padres tienen la opción de intercambiar su información genética, entonces se debe determinar aleatoriamente 2 puntos de corte donde los padres intercambiaran su información genética, el proceso se observa en la Figura 3.12.

Padres										Hijos									
1	3	1	2	2	3	2	0	3	1	1	3	1	1	2	2	3	0	3	1
2	1	1	1	2	2	3	3	3	3	2	1	1	2	2	3	2	3	3	3

1er punto de corte                      2do punto de corte

**Figura 3.12 Cruce de 2 puntos**

*Fuente: Elaboración propia*

Al finalizar la etapa se formó el conjunto de hijos mostrado en la Figura 3.13

Hijo N°1	1	3	1	1	2	2	3	0	3	1
Hijo N°2	2	1	1	2	2	3	2	3	3	3
Hijo N°3	3	1	1	2	2	1	2	0	3	1
Hijo N°4	1	3	1	2	2	3	2	2	3	1
Hijo N°5	1	3	1	2	2	3	2	0	3	1
Hijo N°6	2	1	1	1	2	2	3	3	3	3

*Figura 3.13 Población de hijos - Asignación de grupos*

*Fuente: Elaboración propia*

**Mutación:** Luego de haber obtenido la nueva descendencia, cada uno de los hijos pasa por la etapa de mutación. Esto ocurre con baja probabilidad y para este trabajo se determinó que el valor adecuado es 5%.

Para realizar la mutación se empleó un procedimiento llamado “mutarIndividuos”, este método recorre la nueva población de hijos y dependiendo de una probabilidad generada al azar determina si tal individuo debe mutar alguno de sus genes. Por ejemplo, si el valor generado para el último hijo es menor o igual a 0.05 entonces dicho individuo se mutará y la posición del gen a mutar es un aleatorio entre 0 y el tamaño del cromosoma, el procedimiento seguido se muestra en la Figura 3.14 y luego se reemplaza la antigua población por la población de hijos.



**Tabla 3.2****Asignación de grupos**

<b>Clase</b>	<b>Pos. -Grupo</b>	<b>Grupo</b>
Nº1	3	04
Nº2	1	02
Nº3	1	02
Nº4	2	03
Nº5	3	04
Nº6	0	01
Nº7	3	04
Nº8	0	01
Nº9	2	03
Nº10	0	01

**3.6.3 Asignación de aulas**

En esta etapa se asigna las aulas a las diferentes clases y cada asignación considera el tipo de aula que necesita dicha clase. Las soluciones deben considerar obligatoriamente el cumplimiento de las restricciones referentes a la asignación de aulas tales como RO7, RO8 y RO9; y el procedimiento seguido se detalla a continuación.

**Iniciar población:** Hace uso del el procedimiento “iniciarPoblacionA”, donde los individuos son estructurados a partir de valores generados al azar, estos valores corresponden a la posición de las aulas que sean del tipo que necesita el curso para ser dictado, con esto se garantiza el cumplimiento de RO9. La restricción R08 no será evaluada debido a que el algoritmo debe encontrar la mejor asignación en la segunda etapa y con esto se garantiza que en una aula solo se tiene como máximo un curso y un docente asignado en el mismo grupo.

A cada posición del gen le corresponde una única clase y para asignarle un valor es necesario obtener el tipo de aula que necesita la clase. Una vez determinado el tipo de aula necesario se verifica si existen aulas del mismo tipo, en caso no exista se agrega un valor negativo al gen indicando la inexistencia de un aula con las características que necesita. Si existen aulas del mismo tipo entonces se obtienen las posiciones de dichas aulas en la lista “aulas” y se genera un aleatorio para luego obtener el valor en la posición generada, este procedimiento se puede observar en la Figura 3.15.



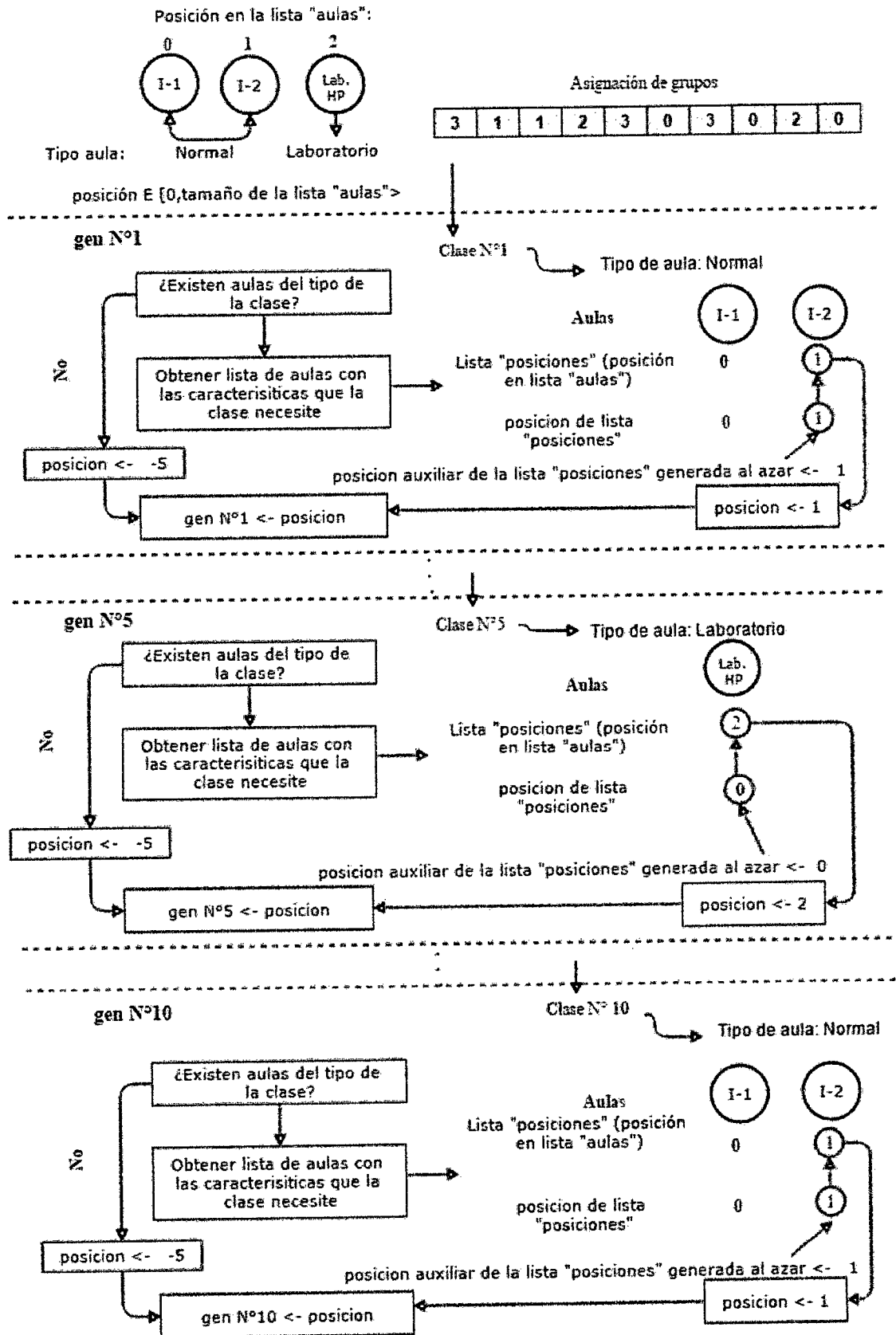


Figura 3.15 Pasos para generar un individuo – Asignación de aulas

Fuente: Elaboración propia

Para seguir con el ejemplo se generó los individuos mostrados en la Figura 3.16.

Indv N°1	0	1	0	0	2	2	1	1	0	0
Indv N°2	0	0	1	1	2	2	0	0	0	1
Indv N°3	1	1	0	0	2	2	1	1	1	1
Indv N°4	0	1	1	1	2	2	0	1	1	0
Indv N°5	0	0	0	1	2	2	0	1	1	1
Indv N°6	1	0	0	1	2	2	0	1	0	1

**Figura 3.16 Población inicial - Asignación de aulas**

*Fuente: Elaboración propia*

**Evaluar la población:** En la etapa tres la única restricción que será evaluada es RO7 y para ello se implementó el procedimiento “evaluarR7xIndv” que es llamado en la función “evaluarPobl ()” antes descrita.

<b>Indv N°4</b>						
	Ubicación de los grupos asignados	Ubicación de las aulas asignadas para Indv N°4				
Ubicación de las clases						*Verificamos que solo 1 clase sea dictada en grupo y aula específica.
1	3	0	1	3	0	
2	1	1	7			
3	1	1	2	1	1	
4	2	1	3			
5	3	2	4	2	1	
6	0	2	9			
7	3	0	5	3	2	
8	0	1	6	0	2	
9	2	1	8	0	1	
10	0	0	10	0	0	
						RO7 = 3

**Figura 3.17 Evaluación de RO7**

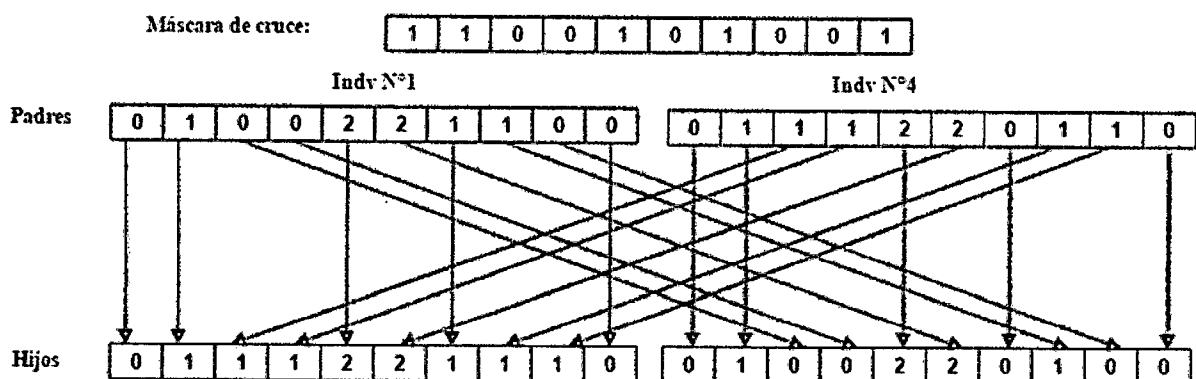
*Fuente: Elaboración propia*

Los individuos de la primera población tuvieron como aptitud 0.9, 0.9, 0.8, 0.7, 0.6 y 0.8 para el primer, segundo, tercero, cuarto, quinto y sexto individuo.

**Verificar condición de término:** Utiliza la función “existeBuenaSol” explicada anteriormente.

**Seleccionar padres:** Hace uso de la función “seleccionarPadres”.

**Cruzar los padres:** Utiliza el método “cruzarPoblacionCruceUnif”, para el ejemplo consideramos que los 2 primeros padres tienen la opción de intercambiar su información genética, entonces para determinar qué padre aportará el primer gen al primer hijo se genera un aleatorio entre 0 y 1; si el valor es 1 entonces el gen del primer descendiente será igual al gen del primer padre, caso contrario será el segundo padre quién aporte el valor del gen; el gen para el segundo hijo será de aquel padre que no aportó su gen al primer hijo. Para realizar este proceso es necesario tener una máscara de cruce cuyo tamaño es igual al tamaño del cromosoma, esta máscara permite determinar que padre será el primero en aportar su material genético. Este proceso es realizado cada vez que se determine el cruce de los padres y su funcionamiento se puede observar en la Figura 3.18.



**Figura 3.18 Cruce Uniforme**

*Fuente: Elaboración propia*

Para el ejemplo seguido se generaron los hijos mostrados en la Figura 3.19.

Hijo N°1	0	1	1	1	2	2	1	1	1	0
Hijo N°2	0	1	0	0	2	2	0	1	0	0
Hijo N°3	0	1	0	0	2	2	0	1	1	1
Hijo N°4	1	0	0	1	2	2	1	1	1	1
Hijo N°5	1	1	0	0	2	2	1	1	1	1
Hijo N°6	1	0	0	1	2	2	0	1	0	1

Figura 3.19 Población de hijos - Asignación de aulas

Fuente: Elaboración propia

**Mutación:** Usa el procedimiento “mutarIndividuos”, cuando se determina la mutación de un individuo el valor del gen a mutar debe ser obtenido siguiendo el mismo procedimiento que el descrito para la etapa de inicialización de la población inicial. En la Figura 3.20 se puede observar la mutación de un gen cuya posición ha sido obtenido al azar.

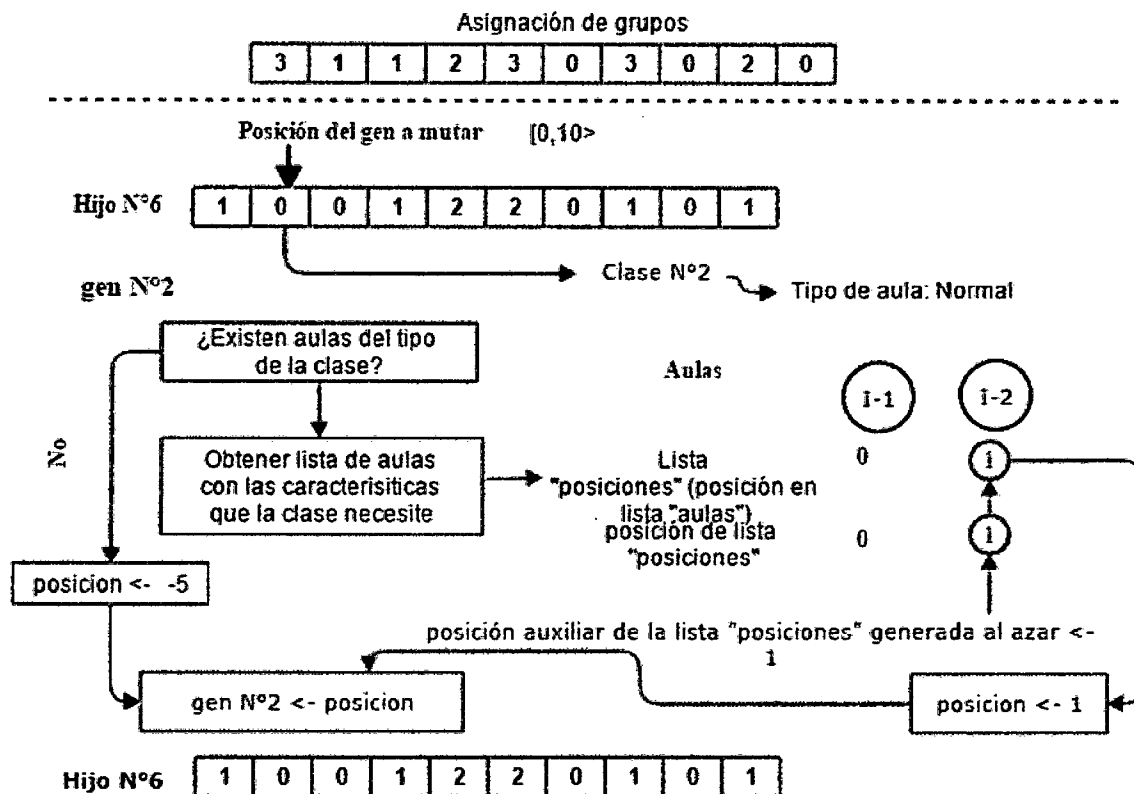


Figura 3.20 Mutación de 1 individuo - Asignación de aulas

Fuente: Elaboración propia

**Elitismo:** Luego de haber reemplazado la población actual por la población de hijos y de haber aplicado mutación se procede a guardar al mejor individuo de la población anterior, para ello se hace uso del procedimiento “elitismo” explicado anteriormente.

A partir de aquí el proceso es iterativo y culmina cuando una de las condiciones de término se cumpla. Luego de haber acabado el proceso se llama al procedimiento “asignarAulas” encargado de asignar el valor del mejor individuo a cada una de las clases, para el ejemplo la asignación quedó con los datos mostrados en la Tabla 3.3.

**Tabla 3.3**

*Asignación de aulas*

<b>Clase</b>	<b>Pos. -Aula</b>	<b>Aula</b>
Nº1	0	I-1
Nº2	1	I-2
Nº3	0	I-1
Nº4	1	I-2
Nº5	2	Lab. Cómputo
Nº6	2	Lab. Cómputo
Nº7	1	I-2
Nº8	1	I-2
Nº9	0	I-1
Nº10	0	I-1

Luego de haber asignado los tres recursos, la programación de clases quedó con los datos mostrados en la Tabla 3.4.

**Tabla 3.4*****Ejemplo de programación de clases***

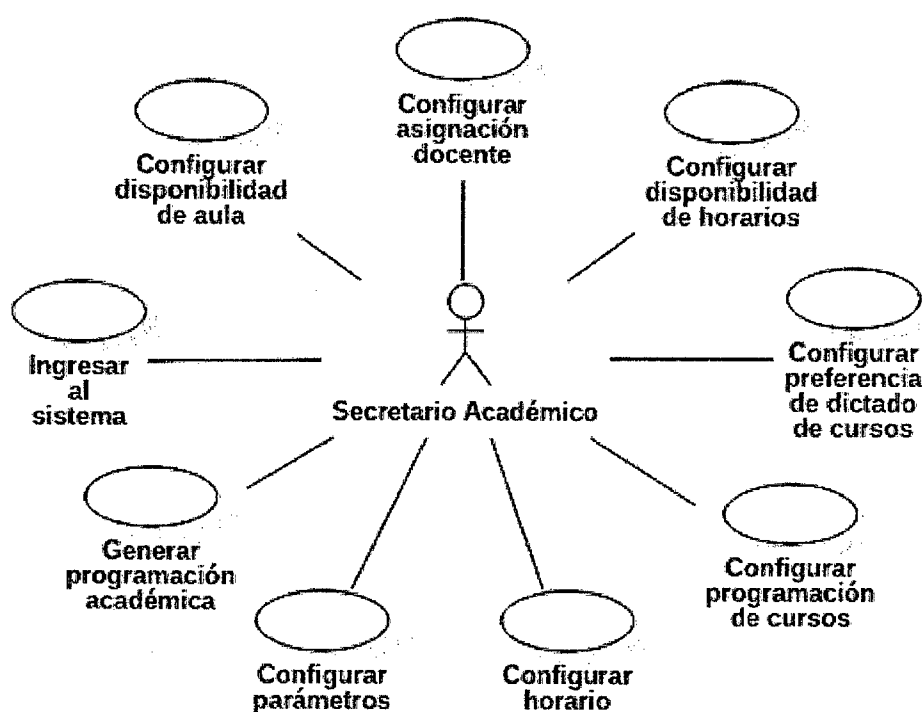
Clase	Curso	Docentes	Grupo	Sección	Aula
Nº1	IO3420	QR0002	04	1	I-1
Nº2	IO3420	QR0002	02	2	I-2
Nº3	IO3412	DD0001	02	1	I-1
Nº4	IO3412	DD0001	03	2	I-2
Nº5	SI3332	-	04	1	Lab. Cómputo
Nº6	SI3332	-	01	2	Lab. Cómputo
Nº7	SI3418	CA0004	04	1	I-2
Nº8	SI3418	-	01	2	I-2
Nº9	SI5492	CA0004	03	1	I-1
Nº10	SI5441	CA0004	01	1	I-1

## CAPÍTULO 4: CONCEPCIÓN, ANÁLISIS Y DISEÑO DEL SISTEMA

### 4.1 CASOS DE USO DEL SISTEMA

Los diagramas de casos de uso nos permiten documentar el comportamiento del sistema desde el punto de vista del usuario, mientras que los casos de uso nos muestran el conjunto de acciones que se realizan como respuesta a un evento que inicia un actor sobre el sistema.

En el diagrama de caso de uso que se muestra en la Figura 4.1 se presentan los requisitos funcionales del sistema, es decir, aquellas funciones que el sistema permite ejecutar; además se presenta al actor que interactuará con el sistema y las asociaciones con los casos de uso.



*Figura 4.1 Diagrama de casos de uso*

*Fuente: Elaboración propia*

#### 4.1.1 Descripción de casos de uso

A continuación se describe lo que sucede con el sistema en cada interacción con el actor, es decir se describe el comportamiento observable del sistema en cada.

#### 4.1.1.1 Caso de uso “Ingresar al sistema”

<b>Actor</b>	Secretario Académico
<b>Pre Condición</b>	El usuario debe estar registrado para que pueda acceder al sistema.
<b>Resumen</b>	La aplicación solo permite el ingreso del Secretario Académico digitando su nombre de usuario y su respectiva contraseña.
<b>Flujo Principal</b>	<ol style="list-style-type: none"><li>1. El actor accede a la aplicación</li><li>2. El sistema muestra en pantalla la interfaz de acceso, donde se solicita un nombre de usuario y contraseña.</li><li>3. El actor digita su nombre de usuario y contraseña.</li><li>4. El actor selecciona la opción “Aceptar”. Ver flujo alternativo 1.</li><li>5. El sistema verifica que la clave de acceso y contraseña sean correctas. Ver flujo alternativo 2 (E1).</li><li>6. Muestra el menú principal del sistema.</li></ol>
<b>Flujo Alternativo</b>	<ol style="list-style-type: none"><li>1. Si el usuario selecciona la opción “Salir”.<ol style="list-style-type: none"><li>1.1 El sistema muestra un mensaje de confirmación, en caso de aceptar el sistema cierra la pantalla de “Acceso de Usuario”, caso contrario continua el paso 2 del flujo principal.</li></ol></li><li>2. Si el actor no completó los campos requeridos (usuario y contraseña).<ol style="list-style-type: none"><li>2.1 El sistema detecta que los datos no se han ingresado y muestra mensaje “Debe ingresar datos requeridos”.</li><li>2.2 El actor selecciona la opción “Aceptar”.</li></ol></li></ol>



	2.3 El sistema cierra el mensaje, continúa el paso 2 del flujo principal.
<b>Excepciones</b>	E1. Si los datos ingresados no son los esperados el sistema mostrará el siguiente mensaje: “Datos ingresados no coinciden”.

#### 4.1.1.2 Caso de uso “Configurar disponibilidad de aula”

<b>Actor</b>	Secretario Académico
<b>Pre Condición</b>	El usuario ingresó satisfactoriamente al sistema.
<b>Resumen</b>	Esta opción del sistema permite cambiar la disponibilidad de las aulas en el actual semestre académico, las aulas cuya disponibilidad estén confirmadas serán usadas en el proceso de elaboración de horarios de clases para dicho semestre.
<b>Flujo Principal</b>	<ol style="list-style-type: none"> <li>1. El actor selecciona “aula” en el menú de configuración.</li> <li>2. El sistema muestra las aulas de la facultad.</li> <li>3. El actor presiona la opción “Modificar”.</li> <li>4. El sistema habilita los campos para cambiar la configuración</li> <li>5. El actor cambia la configuración de las aulas y presiona en “Guardar”. Ver flujo alternativo 1.</li> <li>6. El sistema guarda la configuración de las aulas y muestra un mensaje de configuración exitosa (E1).</li> <li>7. El actor acepta el mensaje de confirmación y cierra la ventana.</li> </ol>
<b>Flujo Alternativo</b>	<ol style="list-style-type: none"> <li>1. Si el actor decide cancelar la modificación <ol style="list-style-type: none"> <li>1.1 El actor presiona el icono “Cancelar”. El sistema deshabilita los campos y muestra la información de las aulas sin haberse realizado los cambios.</li> </ol> </li> </ol>

	1.2 Vuelve al punto 2 del flujo principal.
<b>Excepciones</b>	E1. Si la configuración de aulas no se pudo guardar: “Ocurrió un error inesperado, la configuración no se pudo guardar”.

#### 4.1.1.3 Caso de uso “Configurar asignación docente”

<b>Actor</b>	Secretario Académico
<b>Pre Condición</b>	El Usuario ingresó satisfactoriamente al sistema.
<b>Resumen</b>	Esta opción del sistema permite cambiar la configuración de asignación de los docentes, aquellos docentes cuya asignación está activa se considerará al momento de elaborar los horarios de clases.
<b>Flujo Principal</b>	<ol style="list-style-type: none"> <li>1. El actor selecciona la opción “docente” del menú de configuración.</li> <li>2. El sistema muestra los departamentos académicos suscritos a la FII.</li> <li>3. El actor selecciona el departamento académico y presiona la opción “Buscar”.</li> <li>4. El sistema muestra la lista de docentes suscritos al departamento seleccionado.</li> <li>5. El actor presiona la opción “Habilitar para configurar”.</li> <li>6. El sistema habilita los campos para cambiar la configuración.</li> <li>7. El actor cambia la configuración de los docentes cuyos datos se tendrán en cuenta en la elaboración de horarios y presiona en “Guardar”. Ver flujo alternativo 1.</li> <li>8. El sistema guarda la configuración de las docentes y muestra un mensaje de configuración exitosa (E1).</li> </ol>

	9. El actor acepta el mensaje de confirmación y cierra la ventana.
<b>Flujo</b>	1. Si el usuario decide cancelar la configuración de asignación docente.
<b>Alternativo</b>	1.1 El sistema deshabilita la modificación. 1.2 Vuelve al paso 4 del flujo principal.
<b>Excepciones</b>	E1.Si ocurrió un error en el sistema entonces se mostrará un mensaje “Ocurrió un error inesperado, la configuración no se pudo guardar”.

#### 4.1.1.4 Caso de uso “Configurar disponibilidad de horarios”

<b>Actor</b>	Secretario Académico
<b>Pre Condición</b>	El usuario ingresó satisfactoriamente al sistema.
<b>Resumen</b>	Esta opción del sistema permite cambiar los horarios disponibles de los docentes, estos horarios especifican en que momento los docentes prefieren dictar clases.
<b>Flujo Principal</b>	1. El actor Selecciona la opción “docente” del menú de configuración. 2. El sistema muestra los departamentos académicos suscritos a la FII. 3. El actor selecciona el departamento académico y presiona la opción “Buscar”. 4. El sistema muestra la lista de docentes suscritos al departamento seleccionado. 5. El actor selecciona el docente para modificar sus preferencias y presiona en la opción “Editar preferencias”.

	<ol style="list-style-type: none"> <li>El sistema muestra los datos de la preferencia de horarios del docente si es que lo hubiese, caso contrario muestra el total de los horarios disponibles</li> <li>El actor realiza los cambios que cree necesarios y presiona en “Guardar”. Ver flujo alternativo 1.</li> <li>El sistema guarda la preferencia de horarios del docente y muestra un mensaje de configuración exitosa (E1).</li> <li>El usuario acepta el mensaje de confirmación y cierra la ventana.</li> </ol>
<b>Flujo</b>	<ol style="list-style-type: none"> <li>Si el usuario decide cancelar la configuración de preferencia de horarios.</li> </ol>
<b>Alternativo</b>	<ol style="list-style-type: none"> <li>1.1 El sistema deshabilita la modificación.</li> <li>1.2 Vuelve al paso 4 del flujo principal.</li> </ol>
<b>Excepciones</b>	E1. Si ocurrió un error en el sistema entonces se mostrará un mensaje “Ocurrió un error inesperado, la configuración no se pudo guardar.

#### 4.1.1.5 Caso de uso “Configurar preferencia de dictado de cursos”

<b>Actor</b>	Secretario Académico
<b>Pre Condición</b>	El usuario ingresó satisfactoriamente al sistema.
<b>Resumen</b>	Esta opción del sistema permite cambiar la preferencia que tiene un docente en el dictado de cierto curso.
<b>Flujo Principal</b>	<ol style="list-style-type: none"> <li>El actor Selecciona la opción “docente” del menú de configuración.</li> <li>El sistema muestra los departamentos académicos suscritos a la FII.</li> </ol>

- 
3. El actor selecciona el departamento académico y presiona la opción “Buscar”.
  4. El sistema muestra la lista de docentes suscritos al departamento seleccionado.
  5. El actor selecciona el docente para modificar sus preferencias y presiona en la opción “Editar preferencias”.
  6. El sistema muestra los datos de la preferencia de cursos del docente si es que lo hubiese, caso contrario muestra el total de cursos disponibles dependiendo del departamento académico al que pertenezca. Ver Flujo alternativo 1.
  7. El actor realiza los cambios que cree necesarios y presiona en “Guardar”. Ver flujo alternativo 2.
  8. Guarda la preferencia de horarios del docente y muestra un mensaje de configuración exitosa (E1).
  9. El usuario acepta el mensaje de confirmación y cierra la ventana.
- 

**Flujo**

**Alternativo**

1. Cuando el docente enseña un curso que no pertenece a su mismo departamento.
    - 1.1 El usuario da clic en la opción “Más cursos”
    - 1.2 El sistema muestra una lista de departamentos académicos.
    - 1.3 El usuario selecciona el departamento donde pertenece el curso y da clic en buscar.
    - 1.4 El sistema muestra los cursos que pertenecen al departamento seleccionado.
    - 1.5 El usuario selecciona los cursos.
-

	1.6 Vuelve al paso 6 del flujo principal.
	2. Si el usuario decide cancelar la configuración de preferencia de cursos.
	2.1 El sistema deshabilita la modificación.
	2.2 Vuelve al paso 4 del flujo principal.
<b>Excepciones</b>	E1. Si ocurrió un error en el sistema entonces se mostrará un mensaje “Ocurrió un error inesperado, la configuración no se pudo guardar”.

#### 4.1.1.6 Caso de uso “Configurar programación de cursos”

<b>Actor</b>	Secretario Académico
<b>Pre Condición</b>	El usuario ingresó satisfactoriamente al sistema.
<b>Resumen</b>	Esta opción del sistema permite configurar los cursos que se dictarán, la cantidad de secciones y las escuelas para el cual se abrirá dicho curso.
<b>Flujo Principal</b>	<ol style="list-style-type: none"> <li>1. El actor selecciona la opción cursos del menú de configuración.</li> <li>2. El sistema muestra las diferentes escuelas y sus planes de estudios.</li> <li>3. El actor selecciona la escuela y el plan de estudio y presiona en la opción “Buscar”. Ver flujo alternativo 1.</li> <li>4. El sistema muestra la lista de todos los cursos pertenecientes al plan de estudios seleccionado.</li> <li>5. El usuario selecciona los cursos a configurar y presiona el botón “Configurar”.</li> </ol>

	<hr/> 6. El sistema muestra los datos de los cursos elegidos y habilita campos para modificar. 7. El actor realiza las modificaciones necesarias y presiona en la opción “Guardar”. Ver flujo alternativo 2. 8. El sistema guarda la configuración de cursos y muestra un mensaje de configuración exitosa (E1). 9. El usuario acepta el mensaje de confirmación y cierra la ventana. <hr/>
	1. Si el usuario verifica que el ciclo académico mostrado no corresponde al actual. 1.1 El usuario da clic en el icono “Editar”. 1.2 El sistema habilita la edición del semestre académico. 1.3 El usuario ingresa el semestre y presiona en el icono “Guardar”. 1.4 Vuelve al paso 3 del flujo principal.
<b>Flujo Alternativo</b>	2. Si el actor desea abrir el curso para más de una escuela. 2.1 El actor elige el curso y presiona el botón “Agregar escuela”. 2.2 El sistema muestra una lista de las escuelas. 2.3 El usuario realiza los cambios necesarios. 2.4 Regresa al paso 7 del flujo principal. <hr/>
<b>Excepciones</b>	E1. Si ocurrió un error en el sistema entonces se mostrará un mensaje “Ocurrió un error inesperado, la configuración no se pudo guardar”. <hr/>

#### 4.1.1.7 Caso de uso “Configurar horario”

<b>Actor</b>	Secretario Académico
<b>Pre Condición</b>	El usuario ingresó satisfactoriamente al sistema.
<b>Resumen</b>	Esta opción permite guardar la configuración de los horarios correspondientes a los grupos donde se podrán dictar los cursos en el correspondiente periodo académico.
<b>Flujo Principal</b>	<ol style="list-style-type: none"><li>1. El actor selecciona la opción “horario” del menú de configuración.</li><li>2. El sistema muestra los grupos disponibles.</li><li>3. El actor presiona la opción “Cambiar configuración”.</li><li>4. El sistema habilita los campos para cambiar la configuración.</li><li>5. El actor cambia la configuración de los grupos y presiona en “Guardar”. Ver flujo alternativo 1.</li><li>6. El sistema guarda la configuración de los grupos y muestra un mensaje de configuración exitosa (E1).</li><li>7. El actor acepta el mensaje de confirmación y cierra la ventana.</li></ol>
<b>Flujo Alternativo</b>	<ol style="list-style-type: none"><li>1. Si el usuario decide cancelar la configuración de horarios.<ol style="list-style-type: none"><li>1.1 El sistema deshabilita la modificación.</li><li>1.2 Vuelve al paso 2 del flujo principal.</li></ol></li></ol>
<b>Excepciones</b>	E1. Si ocurrió un error en el sistema entonces se mostrará un mensaje “Ocurrió un error inesperado, la configuración no se pudo guardar”.



#### 4.1.1.8 Caso de uso “Configurar parámetros”

<b>Actor</b>	Secretario Académico
<b>Pre Condición</b>	El usuario ingresó satisfactoriamente al sistema.
<b>Resumen</b>	Esta opción del sistema permite modificar la configuración de los parámetros necesarios para que el AGs trabaje.
<b>Flujo Principal</b>	<ol style="list-style-type: none"><li>1. El actor selecciona la opción “Configurar” del menú programación académica.</li><li>2. El sistema solicita el ingreso de los valores de los parámetros.</li><li>3. El actor ingresa los valores correspondientes y presiona en la opción “Guardar”. Ver flujo alternativo 1.</li><li>4. El sistema guarda los nuevos valores de los parámetros y muestra un mensaje de modificación exitosa (E1).</li><li>5. El actor acepta el mensaje de confirmación y cierra la ventana.</li></ol>
<b>Flujo Alternativo</b>	<ol style="list-style-type: none"><li>1. Si el usuario decide cancelar la configuración de horarios.<ol style="list-style-type: none"><li>1.1 El sistema deshabilita la modificación.</li><li>1.2 Vuelve al paso 2 del flujo principal.</li></ol></li></ol>
<b>Excepciones</b>	E1.Si los datos requeridos no han sido ingresados u ocurrió un error en el sistema entonces se mostrará un mensaje apropiado para cada evento.

#### 4.1.1.9 Caso de uso “Generar programación académica”

<b>Actor</b>	Secretario Académico
<b>Pre Condición</b>	El usuario ingresó satisfactoriamente al sistema.
<b>Resumen</b>	Se debe haber establecido los valores de los parámetros correspondientes a los cursos, docentes, horarios y aulas.
<b>Flujo Principal</b>	<ol style="list-style-type: none"><li>1. El usuario selecciona la opción “Generar” del menú programación académica. Ver flujo alternativo 1.</li><li>2. El sistema muestra en pantalla la interfaz “Generar programación académica”.</li><li>3. El usuario selecciona el icono de la opción “Generar”. Ver flujo alternativo 2 y 3.</li><li>4. El sistema ejecuta el algoritmo genético y habilita la opción “Cancelar” (E1). Ver Flujo alternativo 4.</li><li>5. El usuario selecciona la opción “Guardar”.</li><li>6. El sistema guarda los horarios académicos y muestra mensaje de información.</li><li>7. El usuario acepta el mensaje de confirmación y cierra la ventana</li></ol>
<b>Flujo Alternativo</b>	<ol style="list-style-type: none"><li>1. Si el usuario verifica que el ciclo académico mostrado no corresponde al actual.<ol style="list-style-type: none"><li>1.1 El usuario da clic en el icono “Editar”.</li><li>1.2 El sistema habilita la edición del semestre académico.</li><li>1.3 El usuario ingresa el semestre y presiona en el icono “Guardar”.</li></ol></li></ol>

- 
- 1.4 Vuelve al paso 1 del flujo principal.
  2. Si el sistema detecta que la cantidad de aulas no son suficientes.
    - 2.1 El sistema muestra un mensaje de confirmación
    - 2.2 El usuario cancela la operación y configura las aulas disponibles. Ver flujo alternativo 5.
    - 2.3 Volver al paso 1 del flujo principal o decide presionar en el icono “Cargar” para luego seguir con el paso 3 del flujo principal.
  3. Si el usuario desea cambiar la ubicación de donde se guardarán los reportes.
    - 3.1 El usuario presiona en el icono de la opción para cambiar la ubicación.
    - 3.2 El sistema muestra un selector de archivos.
    - 3.3 El usuario selecciona la ubicación y presiona en abrir.
    - 3.4 El sistema guarda la ruta señalada.
    - 3.5 Vuelve al paso 4 del flujo principal.
  4. Si el usuario decide cancelar la operación.
    - 4.1 El usuario presiona el icono de la opción “Cancelar”.
    - 4.2 El sistema cancela la ejecución del AGs y deshabilita los campos correspondientes.
    - 4.3 Vuelve al paso 2 del flujo principal.
  5. Si el usuario decide continuar con la operación haciendo caso omiso a la advertencia.
-

---

5.1 El usuario acepta el mensaje para continuar con la generación.

5.2 Vuelve al paso 4 del flujo principal.

---

**Excepciones** E1.Si ocurrió un error en el sistema entonces se mostrará un mensaje “Ocurrió un error inesperado, la configuración no se pudo guardar”.

---

## 4.2 DIAGRAMAS DE SECUENCIA

En los diagramas de secuencia presentados a continuación se muestra la participación de los objetos y la secuencia ordenada de los mensajes intercambiados entre ellos con la finalidad de mostrar a detalle cómo interactúan los diferentes componentes del sistema para lograr cumplir con el objetivo de cada caso de uso.

### 4.2.1 Diagrama de secuencia “Ingreso al sistema”

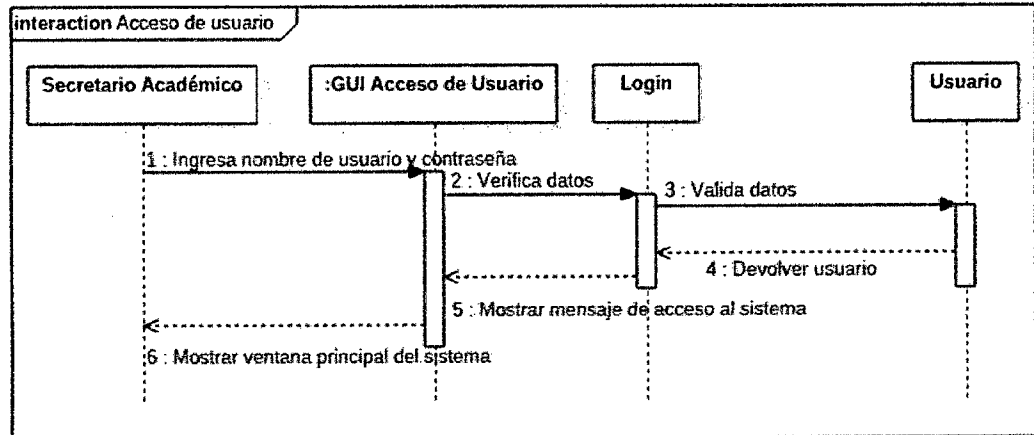


Figura 4.2 Diagrama de secuencia “Ingreso al sistema”

Fuente: Elaboración propia

#### 4.2.2 Diagrama de secuencia “Configurar disponibilidad de aula”

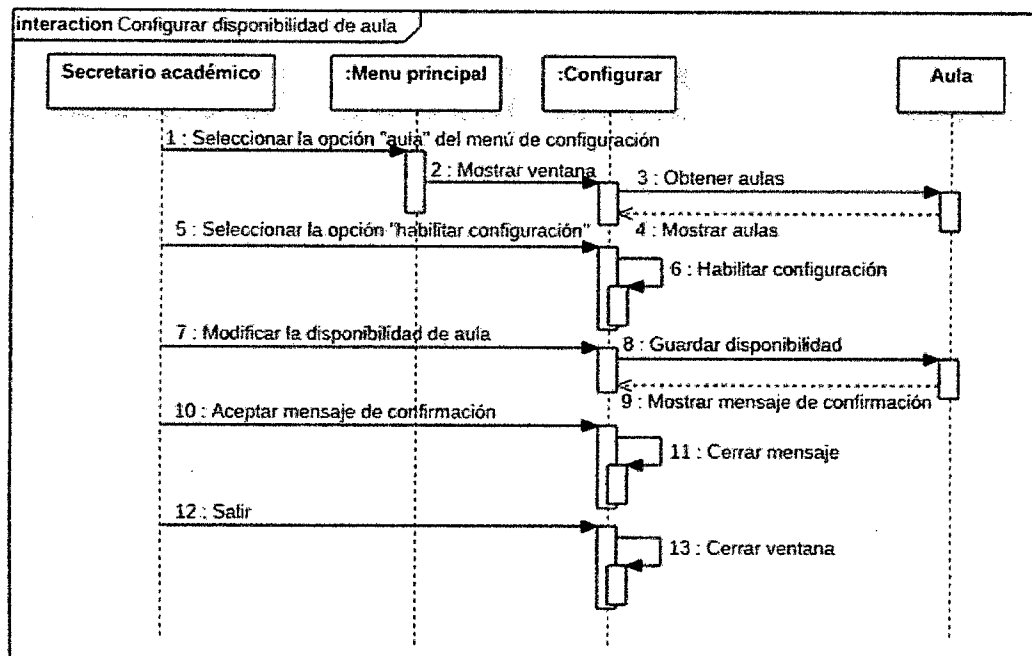


Figura 4.3 Diagrama de secuencia “Configurar disponibilidad de aula”

Fuente: Elaboración propia

#### 4.2.3 Diagrama de secuencia “Configurar asignación docente”

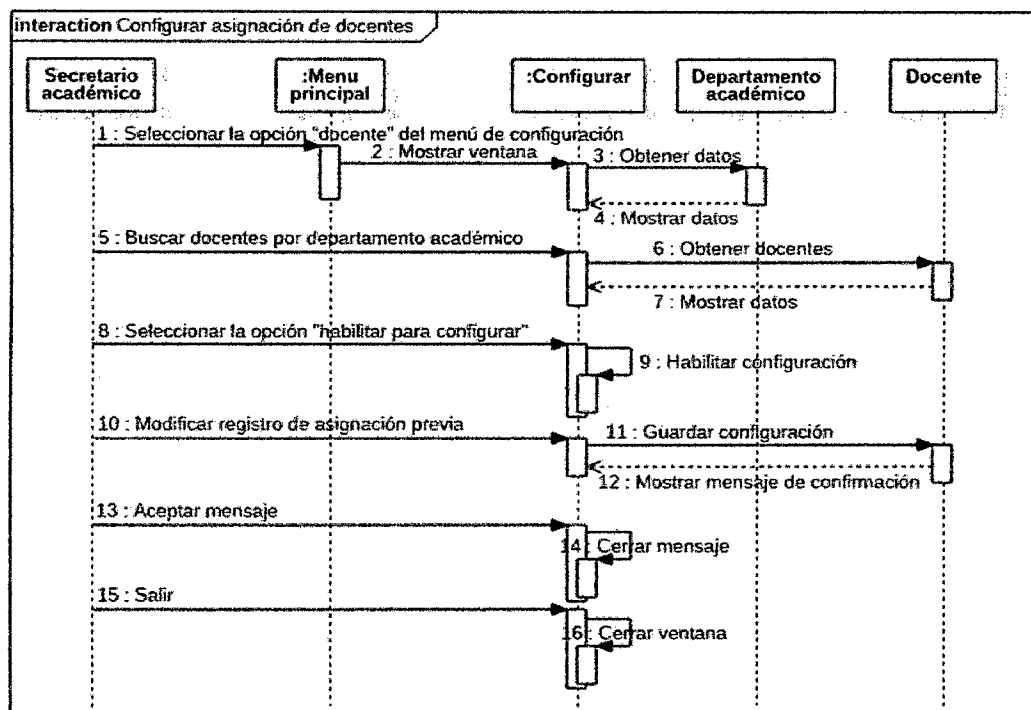
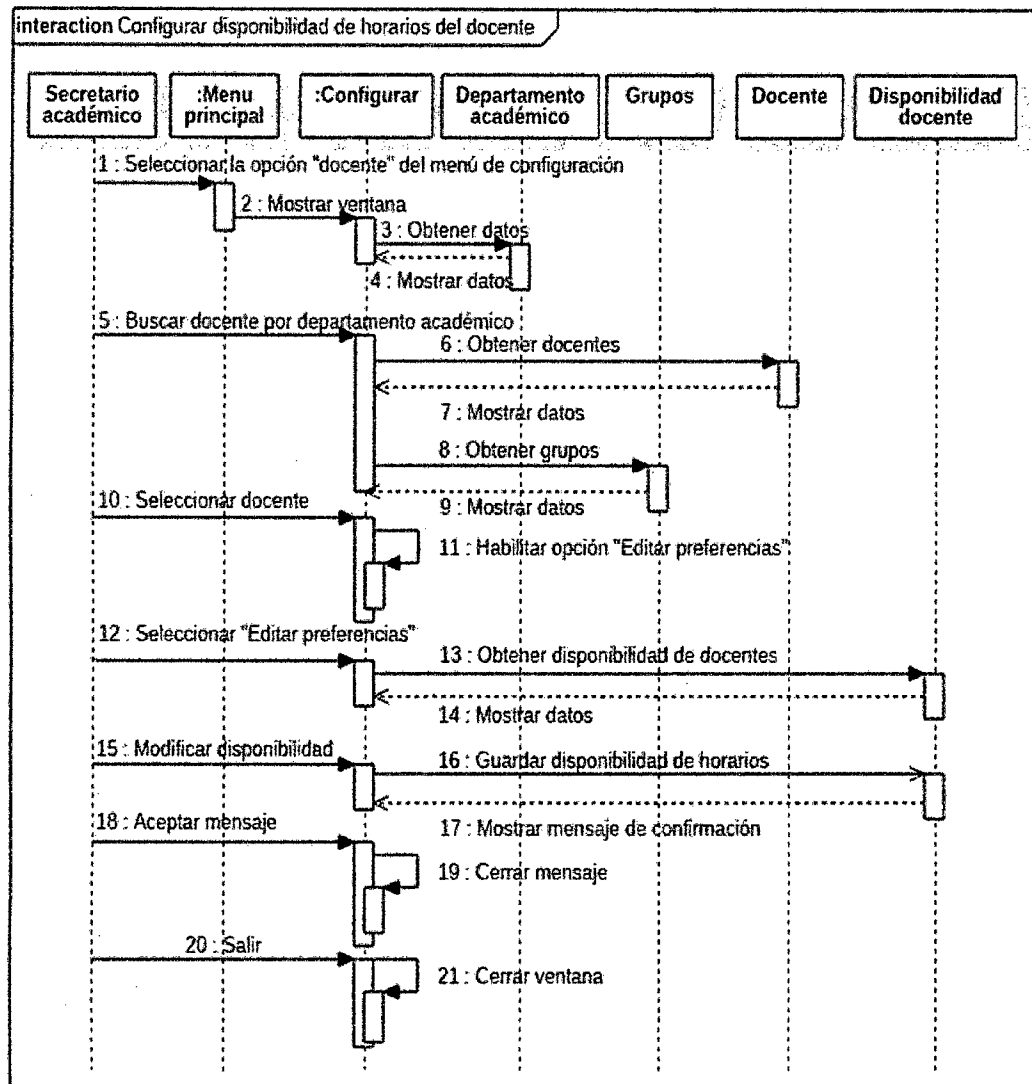


Figura 4.4 Diagrama de secuencia “Configurar asignación docente”

Fuente: Elaboración propia

#### 4.2.4 Diagrama de secuencia “Configurar disponibilidad de horarios”



**Figura 4.5 Diagrama de secuencia “Configurar disponibilidad de horarios”**

*Fuente: Elaboración propia*

#### 4.2.5 Diagrama de secuencia "Configurar preferencia de dictado de cursos"

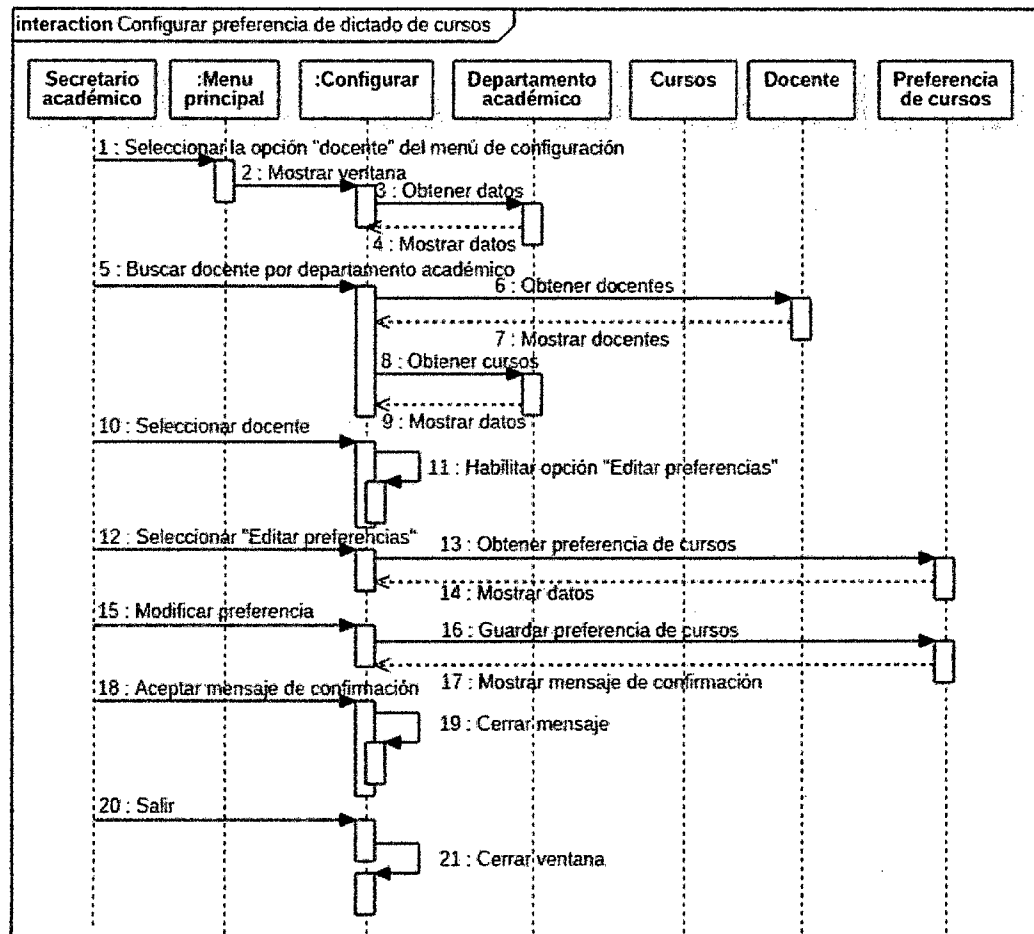
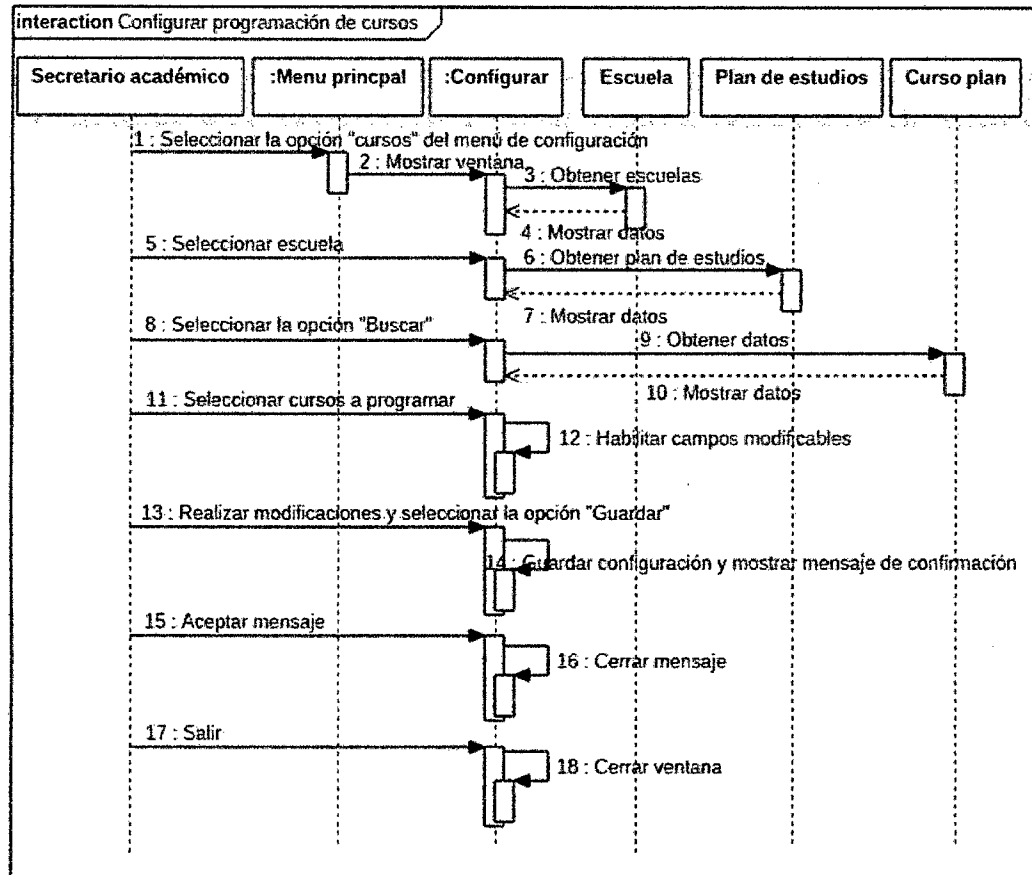


Figura 4.6 Diagrama de secuencia "Configurar preferencia de dictado de cursos"

Fuente: Elaboración propia

#### 4.2.6 Diagrama de secuencia “Configurar programación de cursos”



**Figura 4.7 Diagrama de secuencia “Configurar programación de cursos”**

*Fuente: Elaboración propia*



#### 4.2.7 Diagrama de secuencia “Configurar horario”

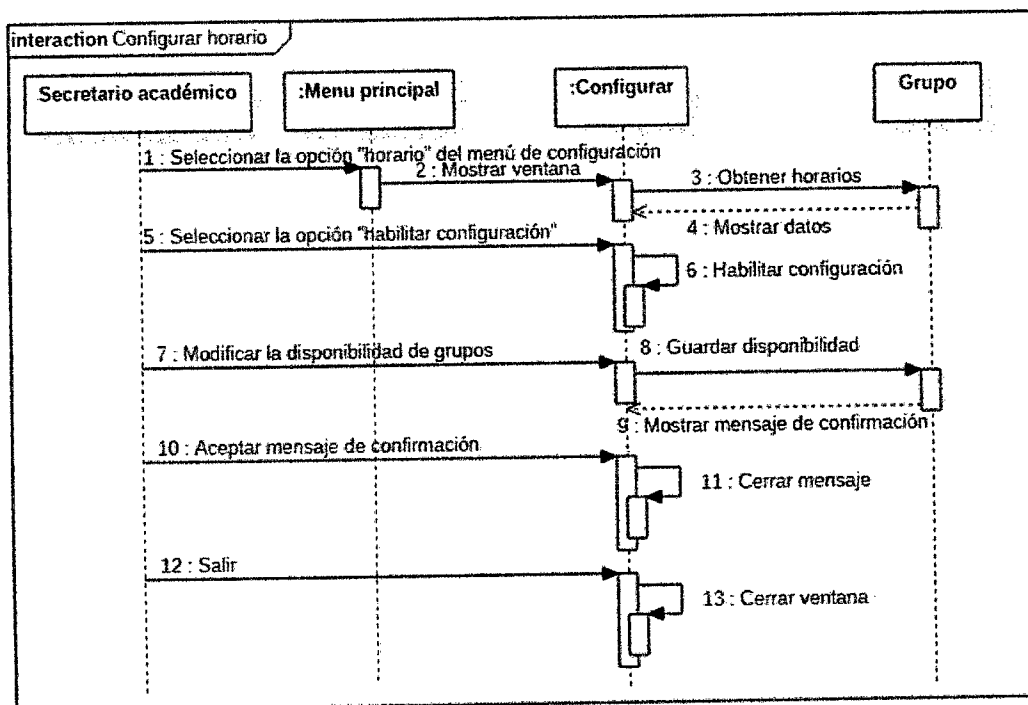


Figura 4.8 Diagrama de secuencia “Configurar horario”

Fuente: Elaboración propia

#### 4.2.8 Diagrama de secuencia “Configurar parámetros”

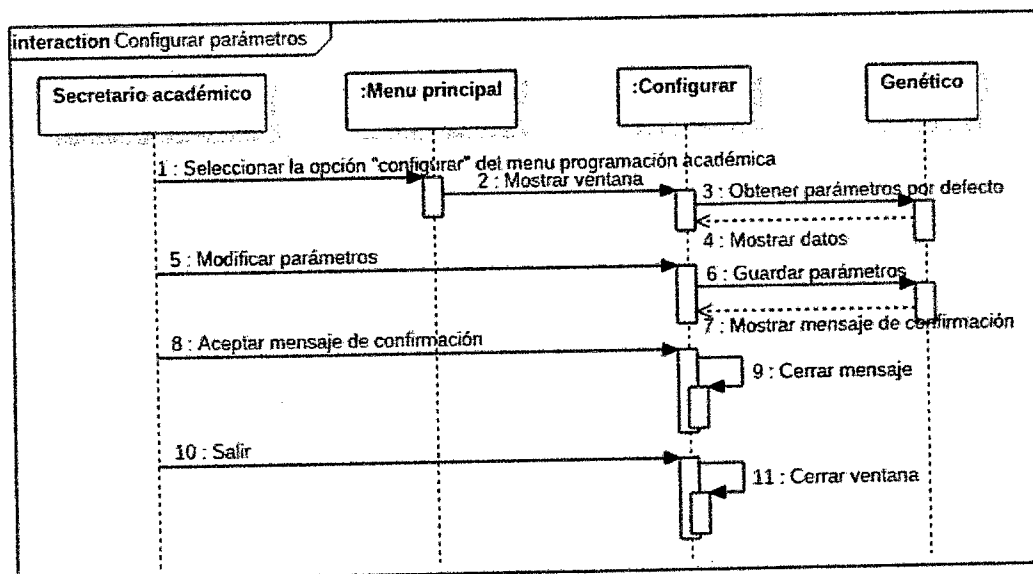
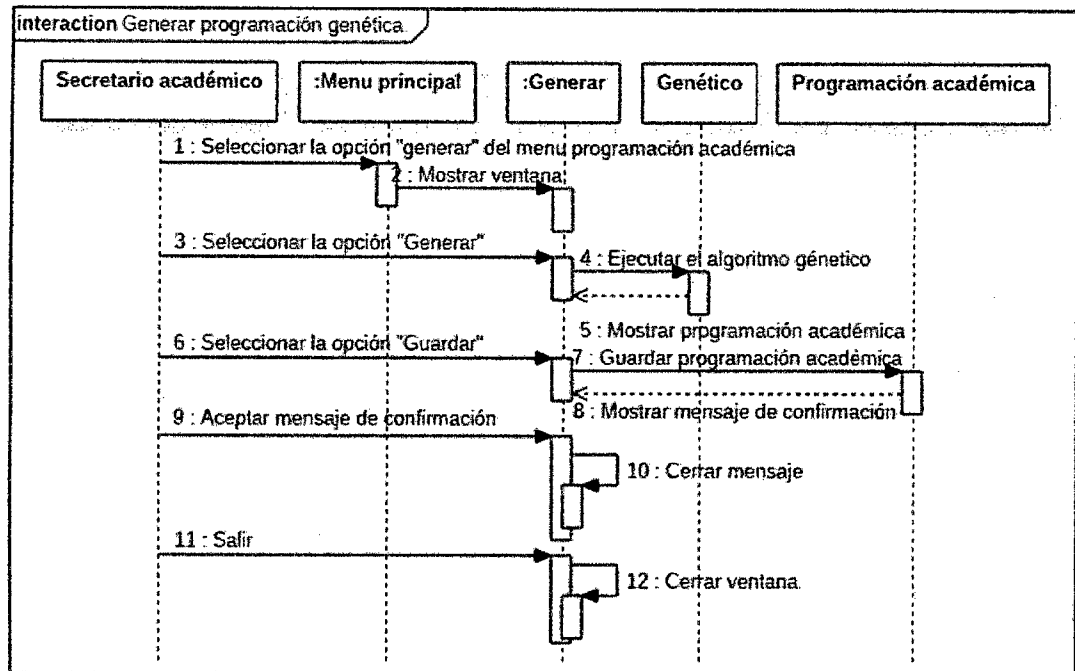


Figura 4.9 Diagrama de secuencia “Configurar parámetros”

Fuente: Elaboración propia

#### 4.2.9 Diagrama de secuencia “Generar programación académica”



**Figura 4.10** Diagrama de secuencia “Generar programación académica”

*Fuente: Elaboración propia*

Diagrama de Base de Datos para el Sistema de Gestión de la Calidad de la Educación Superior (SGCES).

Entidades y sus atributos:

- aula**: codAula INT(11), descripcion VARCHAR(255), codTipoAula INT(11), capacidad INT(11), ubicacion VARCHAR(200), disponible CHAR(1), estado CHAR(1).
- clase**: codClase INT(11), codProgA INT(11), codCurso CHAR(6), codAula INT(11), codDocente CHAR(6), codGrupo CHAR(3), seccion INT(11).
- clase\_detalle**: codClase INT(11), codEscuela INT(11), estado CHAR(1).
- curso**: codCurso CHAR(6), codDepartamento CHAR(2), nombreCurso VARCHAR(300), ht INT(11), hp INT(11), estado CHAR(1).
- escuela**: codEscuela INT(11), codFacultad CHAR(2), correlativo CHAR(1), descripcionEscuela VARCHAR(200), descEsc VARCHAR(100).
- plan**: codPlan INT(11), semestreInicio CHAR(5), codEscuela INT(11), estado CHAR(1).
- curso\_plan**: codCurso CHAR(6), codPlan INT(11), cIdo VARCHAR(2), tipo VARCHAR(1), estado CHAR(1).
- grupo**: codGrupo CHAR(3), descripcionG VARCHAR(50), turno CHAR(1), estado CHAR(1), disponible CHAR(1).
- usuario**: codUsuario CHAR(6), password VARCHAR(33), estado CHAR(1).
- programacion\_academica**: codProgA INT(11), codFacultad CHAR(2), codUsuario CHAR(6), semestre CHAR(5).
- categoria**: codCategoria CHAR(2), descripcion VARCHAR(50), estado CHAR(1), horasMin INT(11).
- preferencia\_docente\_curso**: codDocente CHAR(6), codCurso CHAR(6), estado CHAR(1).
- dedicacion**: codDedicacion CHAR(2), descripcion VARCHAR(20), hrsPermanencia INT(11).
- docente**: codDocente CHAR(6), codDepartamento CHAR(2), codDedicacion CHAR(2), codCategoria CHAR(2), nombres VARCHAR(50), apellidoPaterno VARCHAR(100), apellidoMaterno VARCHAR(100), fechaIngreso DATE, asignar CHAR(1), estado CHAR(1).
- departamento\_academico**: codDepartamento CHAR(2), codFacultad CHAR(2), nombreDepartamento VARCHAR(250), estado CHAR(1).
- disponibilidad\_docente\_grupo**: codDocente CHAR(6), codGrupo CHAR(3), estado CHAR(1).

Relaciones:

- aula** (1) a **clase** (1..\*): 1 a 1..\*
- aula** (1) a **clase\_detalle** (1..\*): 1 a 1..\*
- aula** (1) a **plan** (1..\*): 1 a 1..\*
- aula** (1) a **curso\_plan** (1..\*): 1 a 1..\*
- aula** (1) a **grupo** (1..\*): 1 a 1..\*
- aula** (1) a **usuario** (1..\*): 1 a 1..\*
- aula** (1) a **programacion\_academica** (1..\*): 1 a 1..\*
- aula** (1) a **categoria** (1..\*): 1 a 1..\*
- aula** (1) a **preferencia\_docente\_curso** (1..\*): 1 a 1..\*
- aula** (1) a **dedicacion** (1..\*): 1 a 1..\*
- aula** (1) a **docente** (1..\*): 1 a 1..\*
- aula** (1) a **departamento\_academico** (1..\*): 1 a 1..\*
- aula** (1) a **disponibilidad\_docente\_grupo** (1..\*): 1 a 1..\*
- clase** (1) a **clase\_detalle** (1..\*): 1 a 1..\*
- clase** (1) a **curso** (1..\*): 1 a 1..\*
- clase** (1) a **grupo** (1..\*): 1 a 1..\*
- clase** (1) a **usuario** (1..\*): 1 a 1..\*
- clase** (1) a **programacion\_academica** (1..\*): 1 a 1..\*
- clase** (1) a **categoria** (1..\*): 1 a 1..\*
- clase** (1) a **preferencia\_docente\_curso** (1..\*): 1 a 1..\*
- clase** (1) a **dedicacion** (1..\*): 1 a 1..\*
- clase** (1) a **docente** (1..\*): 1 a 1..\*
- clase** (1) a **departamento\_academico** (1..\*): 1 a 1..\*
- clase** (1) a **disponibilidad\_docente\_grupo** (1..\*): 1 a 1..\*
- clase\_detalle** (1..\*) a **curso** (1..\*): 1..\* a 1..\*
- clase\_detalle** (1..\*) a **grupo** (1..\*): 1..\* a 1..\*
- clase\_detalle** (1..\*) a **usuario** (1..\*): 1..\* a 1..\*
- clase\_detalle** (1..\*) a **programacion\_academica** (1..\*): 1..\* a 1..\*
- clase\_detalle** (1..\*) a **categoria** (1..\*): 1..\* a 1..\*
- clase\_detalle** (1..\*) a **preferencia\_docente\_curso** (1..\*): 1..\* a 1..\*
- clase\_detalle** (1..\*) a **dedicacion** (1..\*): 1..\* a 1..\*
- clase\_detalle** (1..\*) a **docente** (1..\*): 1..\* a 1..\*
- clase\_detalle** (1..\*) a **departamento\_academico** (1..\*): 1..\* a 1..\*
- clase\_detalle** (1..\*) a **disponibilidad\_docente\_grupo** (1..\*): 1..\* a 1..\*
- curso** (1..\*) a **grupo** (1..\*): 1..\* a 1..\*
- curso** (1..\*) a **usuario** (1..\*): 1..\* a 1..\*
- curso** (1..\*) a **programacion\_academica** (1..\*): 1..\* a 1..\*
- curso** (1..\*) a **categoria** (1..\*): 1..\* a 1..\*
- curso** (1..\*) a **preferencia\_docente\_curso** (1..\*): 1..\* a 1..\*
- curso** (1..\*) a **dedicacion** (1..\*): 1..\* a 1..\*
- curso** (1..\*) a **docente** (1..\*): 1..\* a 1..\*
- curso** (1..\*) a **departamento\_academico** (1..\*): 1..\* a 1..\*
- curso** (1..\*) a **disponibilidad\_docente\_grupo** (1..\*): 1..\* a 1..\*
- grupo** (1..\*) a **usuario** (1..\*): 1..\* a 1..\*
- grupo** (1..\*) a **programacion\_academica** (1..\*): 1..\* a 1..\*
- grupo** (1..\*) a **categoria** (1..\*): 1..\* a 1..\*
- grupo** (1..\*) a **preferencia\_docente\_curso** (1..\*): 1..\* a 1..\*
- grupo** (1..\*) a **dedicacion** (1..\*): 1..\* a 1..\*
- grupo** (1..\*) a **docente** (1..\*): 1..\* a 1..\*
- grupo** (1..\*) a **departamento\_academico** (1..\*): 1..\* a 1..\*
- grupo** (1..\*) a **disponibilidad\_docente\_grupo** (1..\*): 1..\* a 1..\*
- usuario** (1..\*) a **programacion\_academica** (1..\*): 1..\* a 1..\*
- usuario** (1..\*) a **categoria** (1..\*): 1..\* a 1..\*
- usuario** (1..\*) a **preferencia\_docente\_curso** (1..\*): 1..\* a 1..\*
- usuario** (1..\*) a **dedicacion** (1..\*): 1..\* a 1..\*
- usuario** (1..\*) a **docente** (1..\*): 1..\* a 1..\*
- usuario** (1..\*) a **departamento\_academico** (1..\*): 1..\* a 1..\*
- usuario** (1..\*) a **disponibilidad\_docente\_grupo** (1..\*): 1..\* a 1..\*
- programacion\_academica** (1..\*) a **categoria** (1..\*): 1..\* a 1..\*
- programacion\_academica** (1..\*) a **preferencia\_docente\_curso** (1..\*): 1..\* a 1..\*
- programacion\_academica** (1..\*) a **dedicacion** (1..\*): 1..\* a 1..\*
- programacion\_academica** (1..\*) a **docente** (1..\*): 1..\* a 1..\*
- programacion\_academica** (1..\*) a **departamento\_academico** (1..\*): 1..\* a 1..\*
- programacion\_academica** (1..\*) a **disponibilidad\_docente\_grupo** (1..\*): 1..\* a 1..\*
- categoria** (1..\*) a **preferencia\_docente\_curso** (1..\*): 1..\* a 1..\*
- categoria** (1..\*) a **dedicacion** (1..\*): 1..\* a 1..\*
- categoria** (1..\*) a **docente** (1..\*): 1..\* a 1..\*
- categoria** (1..\*) a **departamento\_academico** (1..\*): 1..\* a 1..\*
- categoria** (1..\*) a **disponibilidad\_docente\_grupo** (1..\*): 1..\* a 1..\*
- preferencia\_docente\_curso** (1..\*) a **dedicacion** (1..\*): 1..\* a 1..\*
- preferencia\_docente\_curso** (1..\*) a **docente** (1..\*): 1..\* a 1..\*
- preferencia\_docente\_curso** (1..\*) a **departamento\_academico** (1..\*): 1..\* a 1..\*
- preferencia\_docente\_curso** (1..\*) a **disponibilidad\_docente\_grupo** (1..\*): 1..\* a 1..\*
- dedicacion** (1..\*) a **docente** (1..\*): 1..\* a 1..\*
- dedicacion** (1..\*) a **departamento\_academico** (1..\*): 1..\* a 1..\*
- dedicacion** (1..\*) a **disponibilidad\_docente\_grupo** (1..\*): 1..\* a 1..\*
- docente** (1..\*) a **departamento\_academico** (1..\*): 1..\* a 1..\*
- docente** (1..\*) a **disponibilidad\_docente\_grupo** (1..\*): 1..\* a 1..\*
- departamento\_academico** (1..\*) a **disponibilidad\_docente\_grupo** (1..\*): 1..\* a 1..\*

*Fuente: Elaboración propia*

#### 4.4 DIAGRAMA DE COMPONENTES

Cada uno de los componentes mostrados en la Figura 4.11 agrupa un conjunto de clases, el sistema trabaja bajo una arquitectura de capas. La capa de presentación es un componente que a su vez está compuesta por otras clases que permiten mostrar las interfaces del sistema, la capa de negocio agrupa todas aquellas clases que determinan el comportamiento del sistema, algunas de sus clases implementan la interfaz del algoritmo genético y la capa de acceso a datos o capa de persistencia agrupa las clases que permiten ir al SGBD para obtener los datos necesarios que requiere alguna clase ubicada en la capa de negocios.

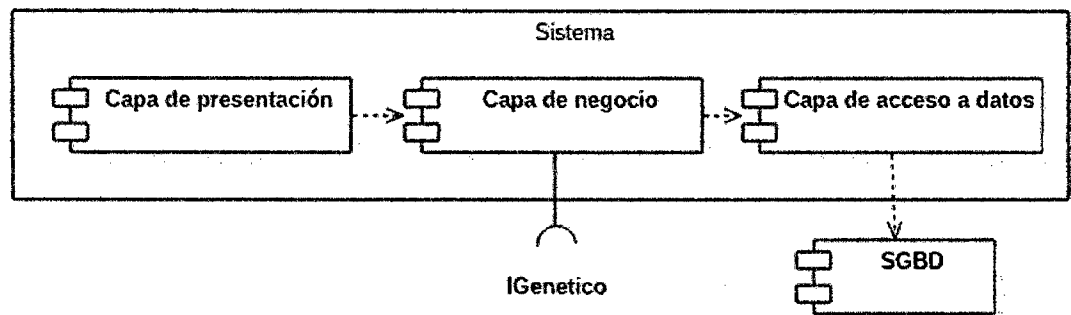


Figura 4.12 Diagrama de componentes

Fuente: Elaboración propia

#### 4.5 DIAGRAMA DE DESPLIEGUE

La arquitectura física del sistema es mostrada en la Figura 4.12, en ella observamos que para la ejecución del sistema solo necesitamos de una máquina que hará de cliente, un servidor de datos y de una red LAN para que puedan comunicarse.

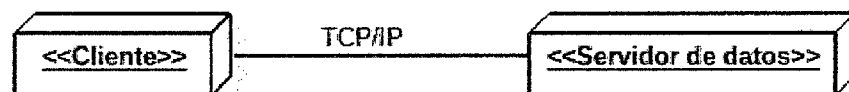


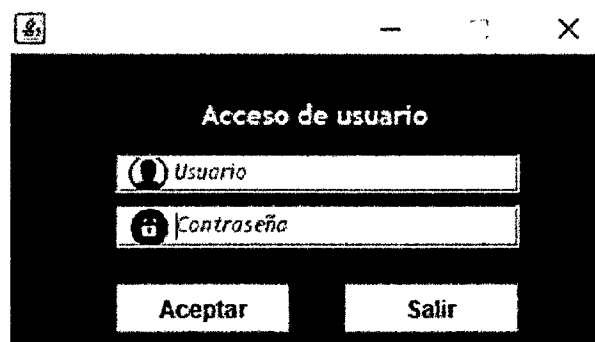
Figura 4.13 Diagrama de despliegue

Fuente: Elaboración propia

## 4.6 INTERFACES DEL SISTEMA

### 4.6.1 Interfaz “Acceso de usuario”

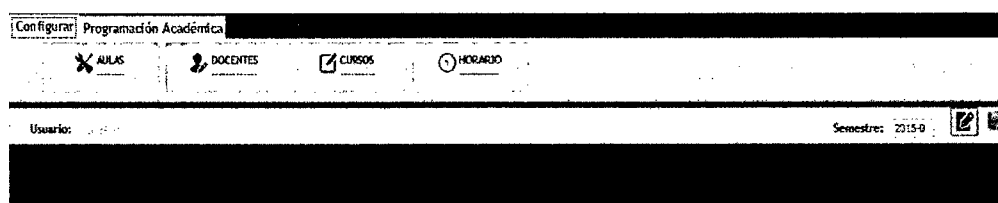
Esta interfaz fue realizada a partir de lo especificado en el caso de uso “Ingreso al sistema” donde el usuario debe colocar sus credenciales de acceso para poder acceder a la aplicación.



*Figura 4.14 Interfaz "Acceso de usuario"*

*Fuente: Elaboración propia*

Una vez que el usuario se ha autenticado se mostrará la pantalla del menú principal donde se muestran las diferentes opciones del sistema.

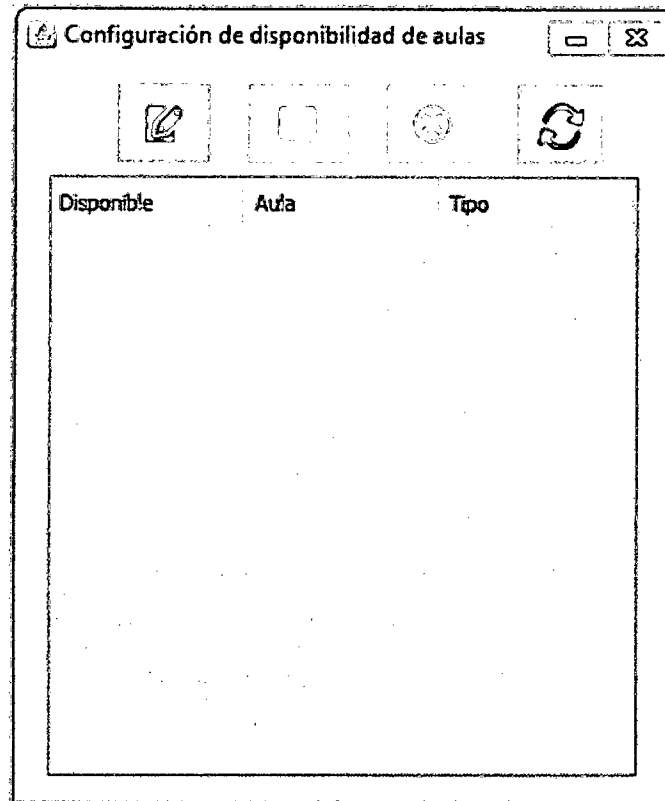


*Figura 4.15 Interfaz "Menú principal"*

*Fuente: Elaboración propia*

### 4.6.2 Interfaz “Configurar disponibilidad de aula”

Esta pantalla es construida a partir del caso de uso “Configurar disponibilidad de aula” donde el usuario puede cambiar la disponibilidad de aulas que se usarán en el semestre para la elaboración de la programación académica.

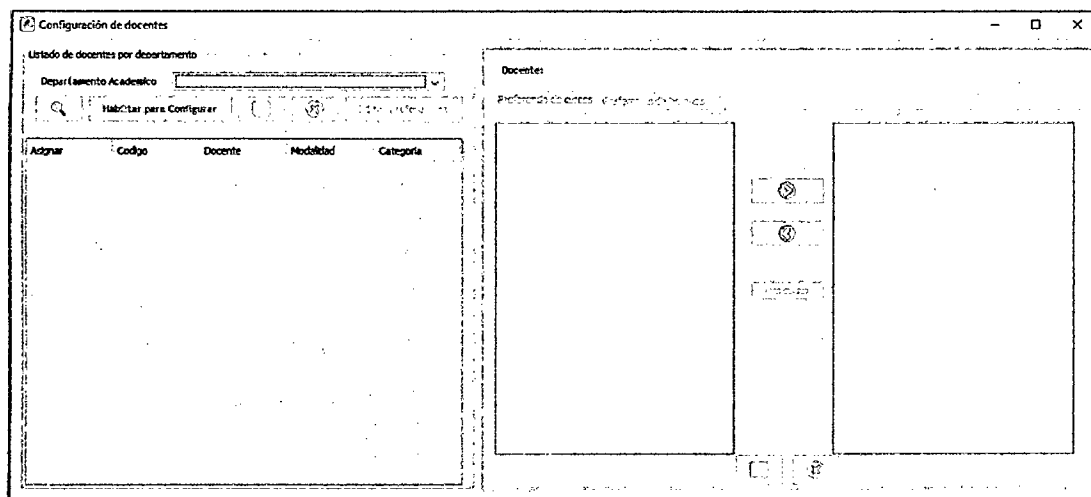


**Figura 4.16** Interfaz "Configurar disponibilidad de aula"

*Fuente: Elaboración propia*

#### 4.6.3 Interfaz "Configuración de docentes"

Esta pantalla es construida a partir de los casos de uso "Configurar asignación docente", "Configurar disponibilidad de horarios" y "Configurar preferencia de dictado de cursos".

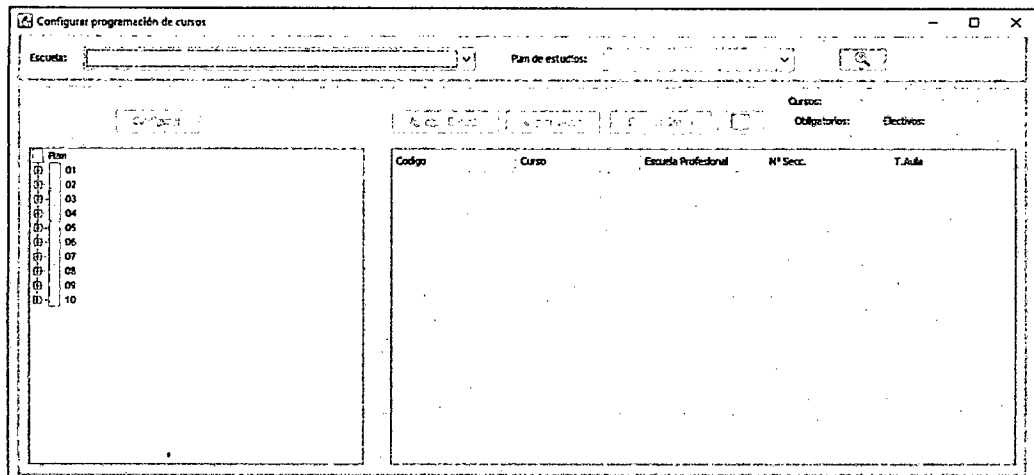


**Figura 4.17** Interfaz "Configuración de asignación docente"

*Fuente: Elaboración propia*

#### 4.6.4 Interfaz “Configurar programación de cursos”

Esta pantalla es construida a partir del caso de uso “Configurar programación de cursos” donde el usuario podrá configurar los cursos y secciones de dichos cursos que se deben dictar en el siguiente semestre académico.

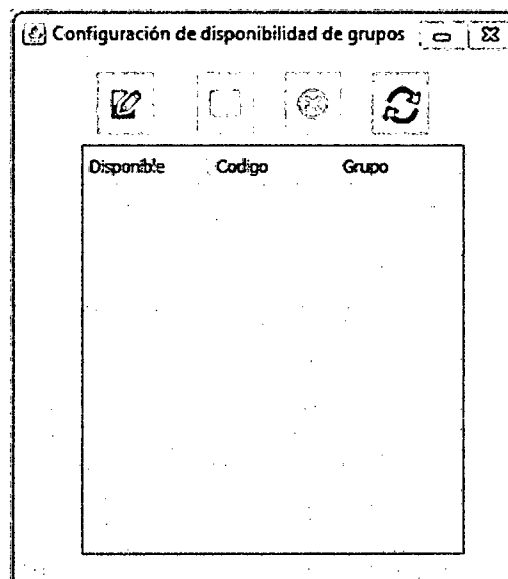


*Figura 4.18 Interfaz "Configurar programación de cursos"*

*Fuente: Elaboración propia*

#### 4.6.5 Interfaz “Configurar horario”

Esta interfaz es construida a partir del caso de uso “Configurar horario”, donde el usuario puede modificar los grupos disponibles para realizar la programación académica.

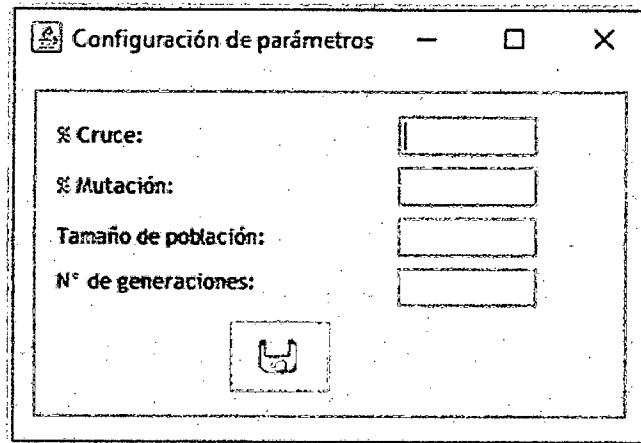


*Figura 4.19 Interfaz "Configurar horario"*

*Fuente: Elaboración propia*

#### 4.6.6 Interfaz “Configurar parámetros”

Esta interfaz es construida a partir del caso de uso “Configurar parámetros”, donde el usuario puede modificar la configuración de los parámetros por defecto para que el AGs trabaje.

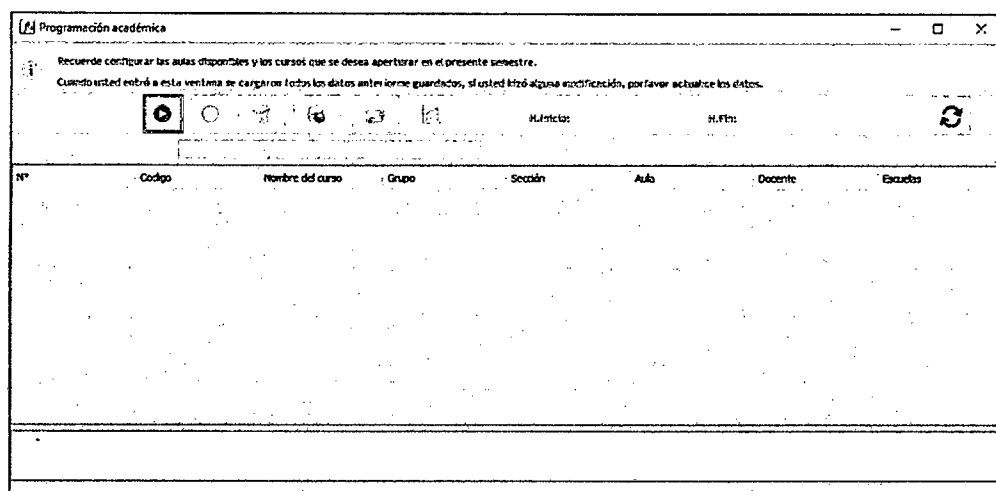


*Figura 4.20 Interfaz "Configurar parámetros"*

Fuente: Elaboración propia

#### 4.6.7 Interfaz “Generar programación académica”

Esta interfaz es construida a partir del caso de uso “Generar programación académica”, donde el usuario puede ejecutar el AGs para generar la programación académica.



*Figura 4.21 Interfaz "Generar programación académica"*

Fuente: Elaboración propia



## **CAPITULO 5: PRUEBAS Y CALIBRACIÓN DEL ALGORITMO**

Luego de haber diseñado el algoritmo genético es necesario determinar qué métodos, de los diferentes operadores, son los adecuados para su funcionamiento y además determinar cuáles son los mejores valores de los parámetros de tal manera que dicha configuración aporte un mejor rendimiento. Para realizar las pruebas se programaron entre 200 y 300 clases, con 52 docentes, 15 y 22 aulas y 14 grupos de horario, luego se ejecutó el algoritmo combinando diferentes valores de los operadores genéticos y parámetros necesarios con la finalidad de determinar con qué valores muestra un mejor comportamiento, dichos valores fueron variados de acuerdo a lo mencionado en el diseño del algoritmo.

Es necesario recordar que las dos últimas etapas hacen uso del AGs y cada una de ellas depende de la anterior, para realizar la calibración se tuvo en cuenta ambas etapas debido al diferente comportamiento que presentaron, necesitando cada una de ellas un promedio diferente de generaciones para encontrar la solución óptima. También cabe señalar que existe una etapa crítica que implica que si al finalizar dicha etapa no se logró encontrar un individuo que represente una solución óptima entonces la siguiente etapa tampoco podrá encontrar a un individuo que represente una buena solución. La etapa de asignación de grupos es considerada la etapa crítica debido a la necesidad de evaluar más restricciones que la etapa de asignación de aulas y además porque si la solución encontrada infringe con la restricción RO6 que considera si la cantidad de clases programadas en un mismo grupo supera la cantidad de aulas del mismo tipo entonces en la siguiente etapa será imposible encontrar la asignación adecuada de aulas.

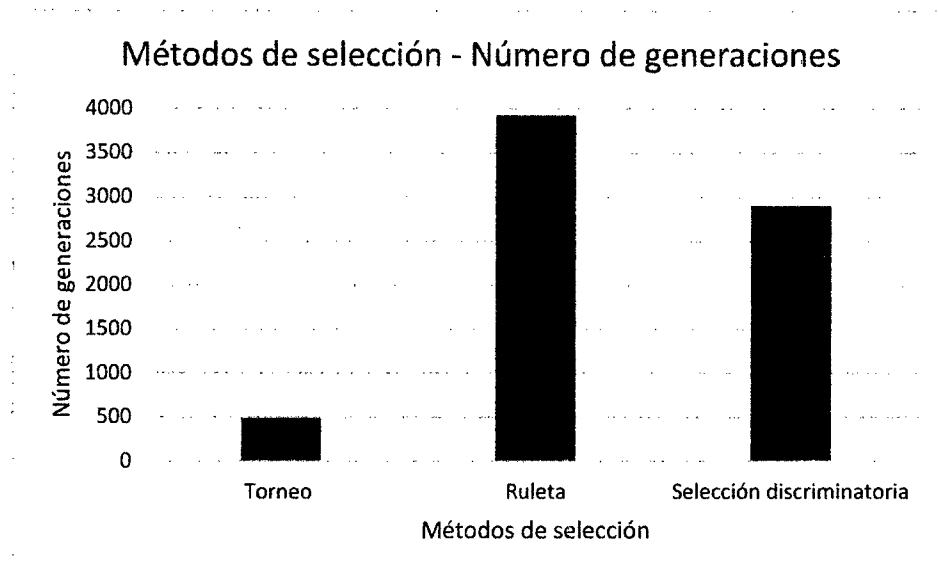
Para la selección de los métodos para los diferentes operadores y para la selección de la configuración de parámetros se tuvo en cuenta dos puntos importantes: el número de generaciones promedio necesarias para encontrar una solución óptima y la frecuencia de éxito.

### **5.1 PRUEBAS SOBRE LOS MÉTODOS DE LOS OPERADORES**

#### **5.1.1 Pruebas sobre los proceso de selección**

Para la elección del método de selección se evaluaron los tres métodos mencionados en el capítulo 3 en función a la cantidad de generaciones hacia la convergencia del algoritmo.

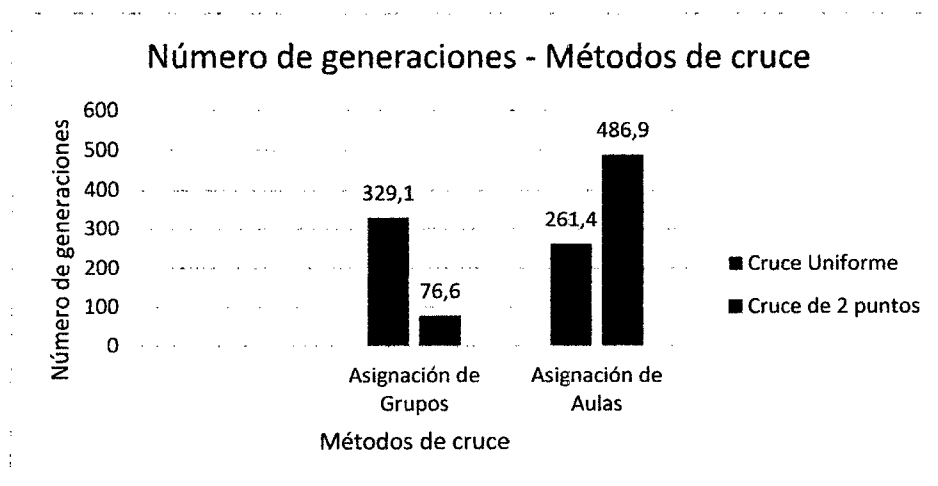
En la Figura 5.1 se muestran los resultados de este comportamiento observándose que la selección por torneo requiere en promedio un menor número de generaciones en comparación a los otros dos métodos.



**Figura 5.1 Métodos de selección vs Número de generaciones**

Fuente: Elaboración propia

### 5.1.2 Pruebas sobre el proceso de cruce



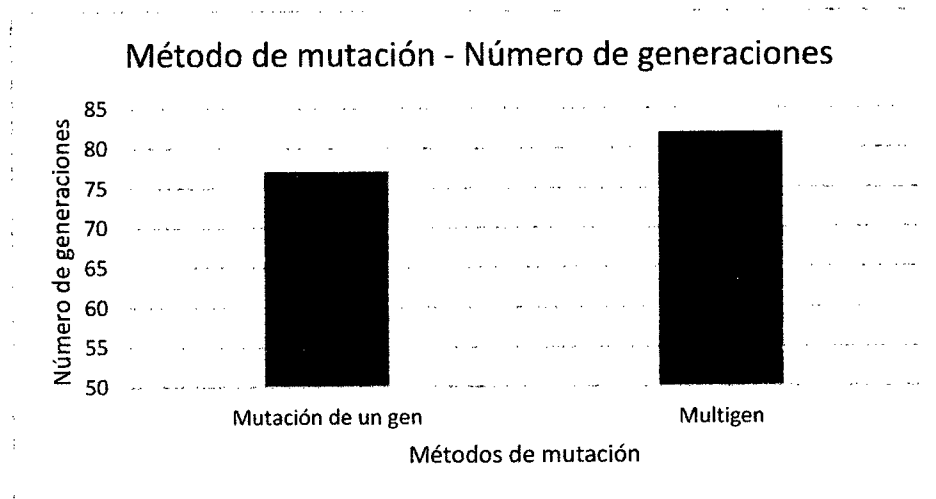
**Figura 5.2 Métodos de cruce vs Número de generaciones**

Fuente: Elaboración propia

En la Figura 5.2 se puede observar el número de generaciones promedio utilizados en las diferentes etapas con los diferentes métodos de cruce. También se puede observar que en la etapa de asignación de grupos el método de cruce de 2

puntos requiere en promedio un menor número de generaciones para encontrar una solución óptima, en cambio en la etapa de asignación de aulas el método de cruce uniforme requiere en promedio un menor número de generaciones que el método de cruce de dos puntos.

### 5.1.3 Pruebas sobre el proceso de mutación



**Figura 5.3 Métodos de mutación vs Número de generaciones**

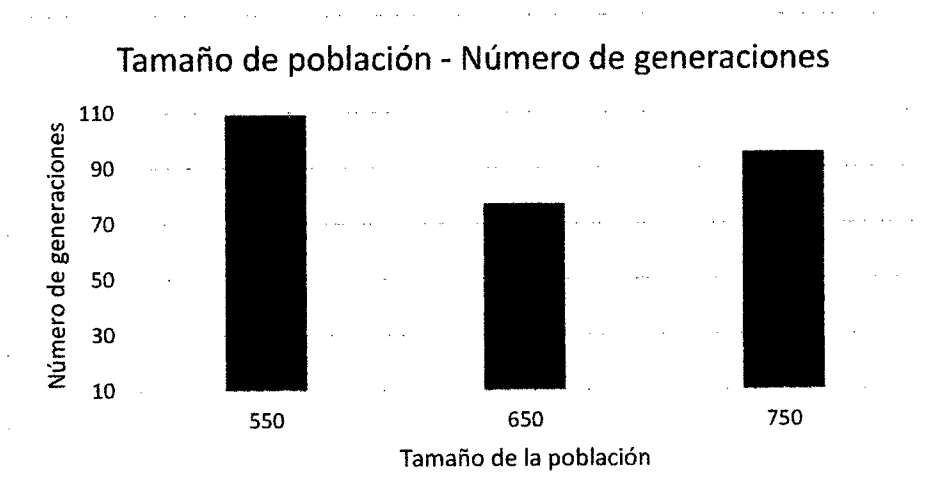
Fuente: Elaboración propia

Los resultados de las pruebas realizadas sobre los métodos de mutación se reflejan en la Figura 5.3 donde se puede observar claramente que el método de mutación de un gen requiere un menor número de generaciones promedio para encontrar una solución óptima.

## 5.2 PRUEBAS SOBRE LOS PARÁMETROS

El objetivo de realizar estas pruebas es conocer qué valores de los parámetros muestran un mejor comportamiento sobre el funcionamiento del algoritmo. Estas pruebas fueron realizadas sobre la etapa crítica mencionada anteriormente.

### 5.2.1 Prueba para el tamaño de población

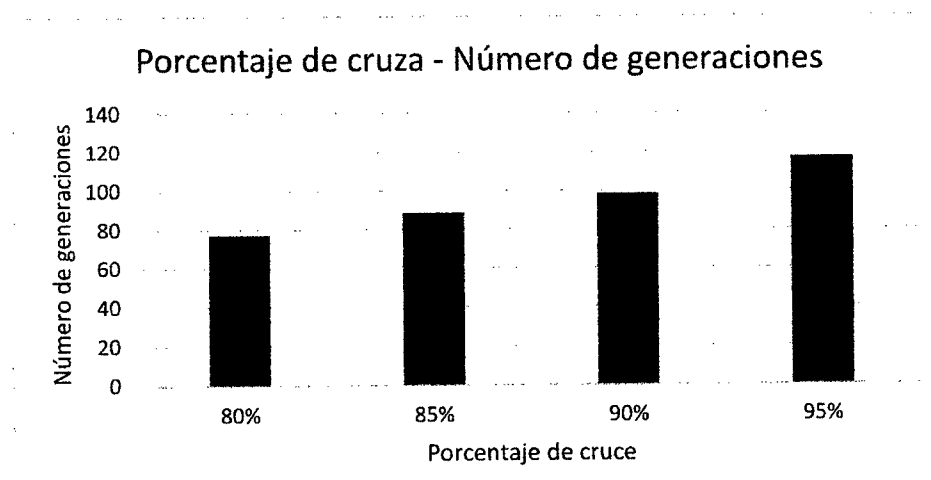


**Figura 5.4** *Tamaño de la población vs. Número de generaciones*

Fuente: Elaboración propia

Como se puede observar en la Figura 5.4 el valor del tamaño de población que permite encontrar la solución óptima en un menor número promedio de generaciones es 650 individuos. Considerando de esta manera que dicho valor es el adecuado para generar la programación académica.

### 5.2.2 Prueba para el porcentaje de cruce

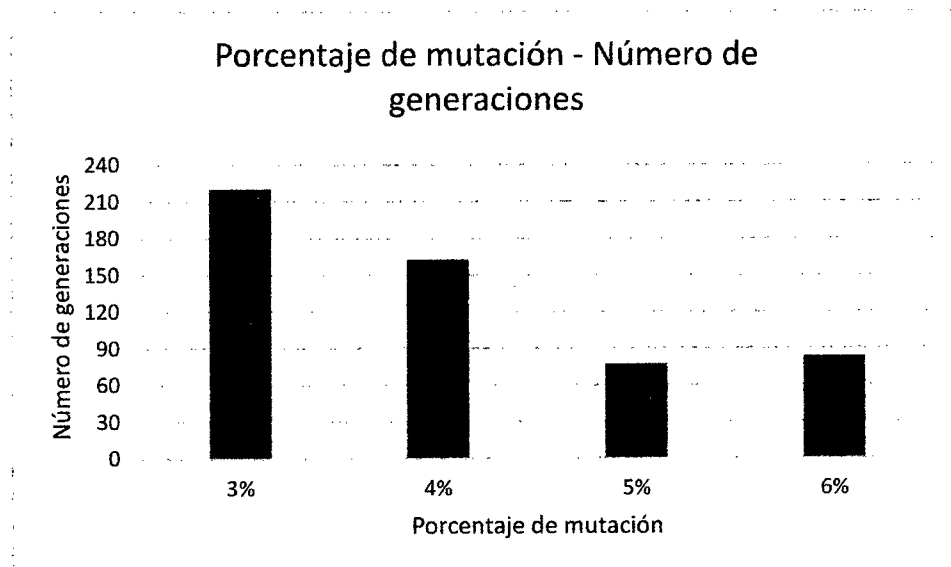


**Figura 5.5** *Porcentaje de cruce vs Número de generaciones*

Fuente: Elaboración propia

En la Figura 5.4 se puede observar el algoritmo genético con un valor de 80% en el porcentaje de cruce requiere de un menor promedio de generaciones para encontrar la solución óptima.

### 5.2.3 Prueba para el porcentaje de mutación



**Figura 5.6 Porcentaje de mutación vs Número de generaciones**

Fuente: Elaboración propia

Como se puede observar en la Figura 5.5 el valor de 5% como porcentaje de cruce permite encontrar soluciones óptimas en un menor número de generaciones.

## CAPÍTULO 6: ANÁLISIS Y DISCUSIÓN DE RESULTADOS

Los resultados muestran que el algoritmo genera soluciones óptimas en todas las ejecuciones, en el Cuadro 6.1 se muestra el Análisis de varianza (ANOVA) donde se observa que la variación de los parámetros % de cruce, % de mutación y tamaño de población no influyen estadísticamente en el número de generaciones hacia la convergencia del algoritmo, comportándose estadísticamente igual. Sin embargo para efectos de elegir el número de porcentaje de cruce, porcentaje de mutación y tamaño de la población, se ha elegido aquellos valores de los parámetros que minimizan el número de generaciones.

**Cuadro 6.1**

*Análisis de varianza para Número de generaciones*

Fuente	Suma de Cuadrados	Gl.	Cuadrado Medio	Razón-F	Valor-P
Efectos principales					
A: Tamaño de población	691,64	2	45,818	0,55	0,5824
B: Porcentaje de cruce	581,95	3	27,318	0,34	0,7952
C: Porcentaje de mutación	66,601	3	88,867	0,12	0,9463
Residuos	0171,8	9	542,87		
Total (corregido)	4012,0	7			

Comentario:

Puesto que ningún valor-P es menor que 0,05; entonces se puede decir que ninguno de los factores tiene un efecto estadísticamente significativo sobre la variable “Número de generaciones” con un 95% de nivel de confianza.

Los operadores genéticos para los cuales el algoritmo funciona eficientemente son: 80% de porcentaje de cruce, 5% de porcentaje de mutación, 650 individuos como tamaño de población y 900 generaciones.

Naupari Quiroz & Rosales Gerónimo (2010) abordaron el mismo problema e implementaron un algoritmo genético para su solución, la diferencia radica en que para este caso todas las etapas no siguen la misma estructura, tampoco se considera las restricciones blandas; lo que hace que el algoritmo propuesto para esta investigación encuentre soluciones

óptimas con más frecuencia, además la configuración de parámetros difieren mucho de los propuestos por ellos.

Cuando se realizaron las pruebas con pocas clases se pudo observar ciertos puntos no tomados en cuenta, uno de ellos fue que la cantidad de clases a programar en un mismo semestre debe ser menor o igual que la cantidad de grupos disponibles. Como la FII cuenta con 52 docentes suscriptos a los diferentes departamentos y la cantidad de aulas disponibles es menor, entonces es necesario definir algunos aspectos que deben ser tomados en cuenta antes de realizar la generación de la programación académica, estos puntos se detallan a continuación:

- Debe existir variedad en los horarios disponibles de los docentes.
- Los docentes que se tendrán en cuenta para la asignación de clases deben de tener una categoría definida.
- Configurar el tipo de aula que requiere una clase específica.

Todas las soluciones obtenidas a través del algoritmo son óptimas y cumple con todas las restricciones obligatorias planteadas en el modelo matemático. Para la generación de la programación es necesario contar con las suficientes aulas para los diferentes cursos y cuando esto no se dé será necesario realizar una previa configuración de aulas ficticias con la finalidad de llevar a cabo la generación y que esta culmine con resultados óptimos ficticios.

Para contrastar la hipótesis planteada se hizo uso de la configuración propuesta y sus resultados son mostrados en la Tabla 6.1, donde se puede observar que el tiempo total para generar la programación académica es mucho menor que el empleado actualmente; concluyendo así que se logró con el objetivo de la investigación.

Tabla 6.1

Resultados de las generaciones

Indicador	Definición operacional	Resultado
Tiempo promedio para configurar la cantidad de aulas disponibles	$tpca = \sum_{i=1}^{ng} \frac{tf - ti}{NG}$ <p> <i>tf</i>: Tiempo final del registro de la configuración de las aulas disponibles.  <i>ti</i>: Tiempo inicial del registro de la configuración de aulas disponibles.  <i>NG</i>: Número de generaciones. </p>	$tpca = \sum_{i=1}^{50} \frac{tf - ti}{50} = 0,5 \text{ minutos}$ <p>Nota: Esta configuración solo es necesario hacerla cuando exista alguna modificación.</p>
Tiempo promedio para configurar los grupos disponibles	$tpcg = \sum_{i=1}^{ng} \frac{tf - ti}{NG}$ <p> <i>tf</i>: Tiempo final del registro de la configuración de grupos disponibles.  <i>ti</i>: Tiempo inicial del registro de la configuración de grupos disponibles.  <i>NG</i>: Número de generaciones. </p>	$tpcg = \sum_{i=1}^{50} \frac{tf - ti}{50} = 0,5 \text{ minutos}$ <p>Nota: Esta configuración solo es necesario hacerla cuando exista alguna modificación.</p>
Tiempo promedio para configurar la asignación de docentes.	$tpcd = \sum_{i=1}^{ng} \frac{tf - ti}{NG}$ <p><i>tf</i>: Tiempo final del registro de la configuración de asignación de docentes.</p>	$tpcd = \sum_{i=1}^{50} \frac{tf - ti}{50} = 4 \text{ minutos}$



	<p><i>ti</i>: Tiempo inicial del registro de la configuración de asignación de docentes.</p> <p><i>NG</i>: Número de generaciones.</p>	<p>Nota: Esta configuración solo es necesario hacerla cuando exista alguna modificación.</p>
<p>Tiempo promedio para configurar la información de los docentes.</p>	<p><math display="block">tpcif = \sum_{i=1}^{ng} \frac{tf - ti}{NG}</math></p> <p><i>tf</i>: Tiempo final del registro de la información de los docentes.</p> <p><i>ti</i>: Tiempo inicial del registro de la información de los docentes.</p> <p><i>NG</i>: Número de generaciones.</p>	<p><math display="block">tpcif = \sum_{i=1}^{50} \frac{tf - ti}{50} = 30 \text{ minutos}</math></p> <p>Nota: Esta configuración solo es necesario hacerla cuando exista alguna modificación.</p>
<p>Tiempo promedio para configurar los valores de los parámetros.</p>	<p><math display="block">tpcp = \sum_{i=1}^{ng} \frac{tf - ti}{NG}</math></p> <p><i>tf</i>: Tiempo final al registrar la configuración de valores de los parámetros.</p> <p><i>ti</i>: Tiempo inicial al registrar la configuración de valores de los parámetros.</p> <p><i>NG</i>: Número de generaciones.</p>	<p><math display="block">tpcp = \sum_{i=1}^{50} \frac{tf - ti}{50} = 0,5 \text{ minutos}</math></p>
<p>Tiempo promedio para configurar los cursos a programar por las</p>	<p><math display="block">tpcc = \sum_{i=1}^{ng} \frac{tf - ti}{NG}</math></p> <p><i>tf</i>: Tiempo final del registro de la configuración de cursos a dictar.</p>	<p><math display="block">tpcc = \sum_{i=1}^{50} \frac{tf - ti}{50} = 10 \text{ minutos}</math></p>

diferentes escuelas.	<p><i>ti</i>: Tiempo inicial del registro de la configuración de cursos a dictar.</p> <p><i>NG</i>: Número de generaciones.</p>	
Tiempo promedio para generar la programación académica.	<p> <math display="block">tpgp = \sum_{i=1}^{ng} \frac{tf - ti}{NG}</math> </p> <p><i>tf</i>: Tiempo final al generar la programación académica.</p> <p><i>ti</i>: Tiempo inicial al generar la programación académica.</p> <p><i>NG</i>: Número de generaciones.</p>	<p> <math display="block">tpgp = \sum_{i=1}^{50} \frac{tf - ti}{50} = 2,5 \text{ minutos}</math> </p>
Total:		48 minutos

## CONCLUSIONES

1. Analizar el proceso de elaboración de la programación académica permitió identificar que el problema podía ser abordado en diferentes etapas de asignación de recursos y además el procedimiento de asignación es diferente en cada una de ellas, por tanto se identificó que la etapa de asignación de docentes no debe hacer uso del algoritmo genético debido a que el procedimiento realizado para su asignación se hace de acuerdo a un orden donde no existe aleatoriedad como en las demás etapas que hacen uso del algoritmo genético.
2. Las etapas de asignación de horarios y asignación de aulas necesitaron usar diferentes métodos de cruce debido al diferente comportamiento mostrado para cada uno de los métodos.
3. Una correcta configuración de los horarios de los docentes permitió que las todas las ejecuciones del sistema culminen con soluciones óptimas. Por tanto se considera que los parámetros % de cruce, % de mutación, número de generaciones y tamaño de población no influyen en la frecuencia de éxito para encontrar soluciones óptimas. Luego se realizó pruebas para determinar la influencia de los parámetros % de cruce, % de mutación y tamaño de población en el número de generaciones donde el análisis de varianza demostró que dichos parámetros no tienen influencia estadística en el número de generaciones necesarias para encontrar una solución óptima.
4. Los valores de los parámetros que se ajustaron al problema fueron 80% de porcentaje de cruce, 5% de porcentaje de mutación, 650 individuos y 900 generaciones. Se ha verificado que valores superiores a los parámetros no contribuyen a una mejora en el comportamiento de la solución.
5. Se desarrolló un sistema que permite automatizar el proceso de la generación de la programación académica, dicho sistema hace uso del algoritmo genético propuesto encontrando soluciones óptimas que cumplen con las restricciones planteadas en el modelo matemático, demostrándose así la hipótesis planteada. Las soluciones son encontradas con un tiempo de ejecución del algoritmo de 2,5 minutos aproximadamente.

## **RECOMENDACIONES**

1. Realizar la conexión con el sistema gestor de base de datos (SGBD) que tiene la Universidad Nacional de Piura con la finalidad de obtener data actualizada, calcular automáticamente la cantidad de secciones que se deben abrir por curso y además para poder guardar la programación académica directamente.
2. Seguir con los procedimientos establecidos a fin de no tener inconveniente al realizar la programación académica.
3. Hacer extensivo la implementación de este algoritmo en todas las facultades de la Universidad Nacional de Piura.
4. Realizar una investigación que incluya la capacidad de aulas como una restricción de asignación.

## BIBLIOGRAFÍA

- Alfonso Galapienso, M. I., Cazorla Quevedo, M. A., Colomina Pardo, O., Escolano Ruiz, F., & Lozano Ortega, M. (2003). Inteligencia artificial: modelos, técnicas y áreas de aplicación. Paraninfo. Obtenido de [http://books.google.com.pe/books?id=\\_spC6S7UfZgC&dq=codificacion+de+variables+en+algoritmos+geneticos&hl=es&source=gbs\\_navlinks\\_s](http://books.google.com.pe/books?id=_spC6S7UfZgC&dq=codificacion+de+variables+en+algoritmos+geneticos&hl=es&source=gbs_navlinks_s)
- Apaza, V. M. A. (2013). Modelo de un Algoritmo Genético con Selección Discriminatoria de Individuos bajo un Esquema de Ponderación de Probabilidades de Mutación.
- Arranz de la Peña, J., & Parra Truyol, A. (s.f.). Algoritmos Genéticos. Obtenido de <http://www.it.uc3m.es/jvillena/irc/practicas/06-07/05.pdf>
- Caballero Fernández, R., Gómez Nuñez, T., Luque Gallego, M., Molina Luque, J., & Torrico González, A. (s.f.). Algoritmos genéticos para la resolución de problemas de Programación por. Obtenido de <http://www.uv.es/asepuma/X/J01C.pdf>
- Centers of Medicare & Medicaid Services. (27 de March de 2008). Selecting a Development Approach. Obtenido de <https://www.cms.gov/Research-Statistics-Data-and-Systems/CMS-Information-Technology/XLC/Downloads/SelectingDevelopmentApproach.pdf>
- Gallego, M. L., González, A. T., & Nuñez, T. G. Algoritmos genéticos para la resolución de problemas de Programación por Metas Entera. Aplicación a la Economía de la Educación. Rafael Caballero Fernández. Julián Molina Luque.
- García Peña., J. F., Ferrán M, A., & Beltran M, M. (19 de 02 de 2013). Obtenido de [http://www.uelbosque.edu.co/sites/default/files/publicaciones/revistas/revista\\_tecnologia/volumen9\\_numero2/analisis\\_complejidad\\_algoritmo9-2.pdf](http://www.uelbosque.edu.co/sites/default/files/publicaciones/revistas/revista_tecnologia/volumen9_numero2/analisis_complejidad_algoritmo9-2.pdf)
- Guerra Cubillos, M. A., Pardo Quiroga, E. H., & Salas Ruiz, R. E. (02 de 07 de 2013). Problema del School Timetabling y algoritmos genéticos: Una revisión. Vínculos, 10(2), 259-276. Obtenido de <http://comunidad.udistrital.edu.co/revistavinculos/files/2013/09/Problema-del-School-Timetabling-y-algoritmos-geneticos.pdf>

Maroto Álvarez, C., Alcaraz Soria, J., & Ruiz García, R. (2002). Investigación operativa: modelos y técnicas de optimización. (2. Ed. Univ. Politéc. Valencia, Ed.) Obtenido de [http://books.google.com.pe/books?id=pqkQPu7jhV0C&pg=PA375&dq=algoritmos+geneticos&hl=es&sa=X&ei=Qv4\\_VK7HCc\\_GgwTqwIDwCg](http://books.google.com.pe/books?id=pqkQPu7jhV0C&pg=PA375&dq=algoritmos+geneticos&hl=es&sa=X&ei=Qv4_VK7HCc_GgwTqwIDwCg)

Castañeda Ramírez, C. H., García Sánchez, E., Martínez Ruiz, F. J., Muñoz Arteaga, J., & (s.f.). Timetabling Académico usando algoritmos genéticos y programación celular. Obtenido de <http://ingsw.ccbas.uaa.mx/sitio/images/pdfpublicaciones/artiCoNaCiCo05-20.pdf>

Johnston Barrientos, J. D. (2000). Aplicación de algoritmos genéticos para la asignación de carga académica en instituciones de educación superior. San Nicolás de los Garza, N.L

Naupari Quiroz, R., & Rosales Gerónimo, G. (2010). Aplicación de algoritmos genéticos para el diseño de un sistema de apoyo a la generación de horarios de clases para la Facultad de Ingeniería de Sistemas e Informática de la UNMSM. Obtenido de <https://www.scribd.com/doc/185547957/Aplicacion-de-Algoritmos-Geneticos-para-el-Diseño-de-un-Sistema-de-Apoyo-a-la-Generación-de-Horarios-de-Clases>

Pacheco Agüero, C. L. (Agosto de 2000). Distribución Óptima de Horarios de Clases utilizando la técnica de Algoritmos Genéticos. Acatlima, Huajuapán de León, Oaxaca.

Pressman, R. S. (1997). Ingeniería del software: un enfoque práctico. (Mikel Angoar, Ed., & R. Ojeda Martín, Trad.) Obtenido de [http://books.google.com.pe/books?id=8UV5jxkuBZIC&hl=es&source=gbs\\_navlinks\\_s](http://books.google.com.pe/books?id=8UV5jxkuBZIC&hl=es&source=gbs_navlinks_s)

Universidad de Tarapacá ARICA-CHILE. (2000). Revista Facultad de Ingeniería. 7(ISSN 0717-1072), 74. Obtenido de [http://books.google.com.pe/books?id=-0hWCiukoYQC&printsec=frontcover&hl=es&source=gbs\\_ge\\_summary\\_r&cad=0#v=onepage&q&f=false](http://books.google.com.pe/books?id=-0hWCiukoYQC&printsec=frontcover&hl=es&source=gbs_ge_summary_r&cad=0#v=onepage&q&f=false)

Valencia, E. (1997, August). Optimización mediante algoritmos genéticos. In Anales del Instituto de Ingenieros de Chile (Vol. 109, No. 2, pp. 83-92).

## **ANEXOS**

### **MANUAL DEL SISTEMA**

#### **1. MANUAL DE INSTALACIÓN DEL SOFTWARE**

##### **1.1 ESPECIFICACIONES DE HARDWARE**

Para la instalación del software es necesario disponer de los siguientes dispositivos:

<b>DISPOSITIVO</b>	<b>CANTIDAD</b>
COMPUTADORA	1
SERVIDOR DE DATOS	1
IMPRESORA	1

Nota: Se recomienda el uso de una computadora con procesador Intel core I3 como mínimo para el buen funcionamiento del sistema.

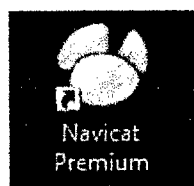
##### **1.2 ESPECIFICACIONES DEL SOFTWARE A UTILIZAR**

Para la instalación del software es necesario disponer de las siguientes herramientas:

- a. Para el servidor de datos:
  - MySQL como SGBD
  - Navicat o cualquier administrador gráfico que facilite la configuración.
- b. Para la maquina cliente:
  - JDK y JRE versión  $\geq 1.8$

##### **1.3 ESPECIFICACIONES DE CONFIGURACIÓN DEL SERVIDOR DE DATOS**

1. Abrir la herramienta de administración.
2. Hacer doble clic en el icono de la herramienta que usaremos, tal como se muestra en la Figura 1.



**Figura 1** Icono del administrador del SGBD

*Nota: Fuente propia*

3. Crear un usuario de base de datos y asignar los permisos necesarios.

Una vez cargado el entorno gráfico es necesario conectarse al SGBD con las credenciales proporcionadas por el administrador.

Se debe crear un usuario y asignarle los permisos necesarios para poder crear esquemas. El script para la creación y asignación es el siguiente:

```
CREATE USER 'p_acad'@'localhost' IDENTIFIED BY 'pa.tesis';  
GRANT SELECT, INSERT, UPDATE, REFERENCES, CREATE, DROP,  
DELETE, ALTER, CREATE VIEW, SHOW VIEW, EXECUTE, CREATE  
ROUTINE, ALTER ROUTINE ON 'prog\_acad'.* TO 'p_acad'@'localhost';
```

4. Crear una nueva conexión

Si creó el usuario con los datos proporcionados en el script anterior entonces necesita ingresar los datos que se muestran en la Figura 2.

La imagen muestra la ventana de configuración de conexión de Navicat Premium. En la parte superior hay una pestaña "General" seleccionada, con otras pestañas "Advanced", "SSL", "SSH" y "HTTP". Debajo, hay campos para configurar la conexión: "Connection Name" con el valor "tesis", "Host Name/IP Address" con "localhost", "Port" con "3306", "User Name" con "p\_acad", y "Password" con caracteres ocultos por puntos. Al final, hay una casilla de verificación "Save Password" que está marcada.

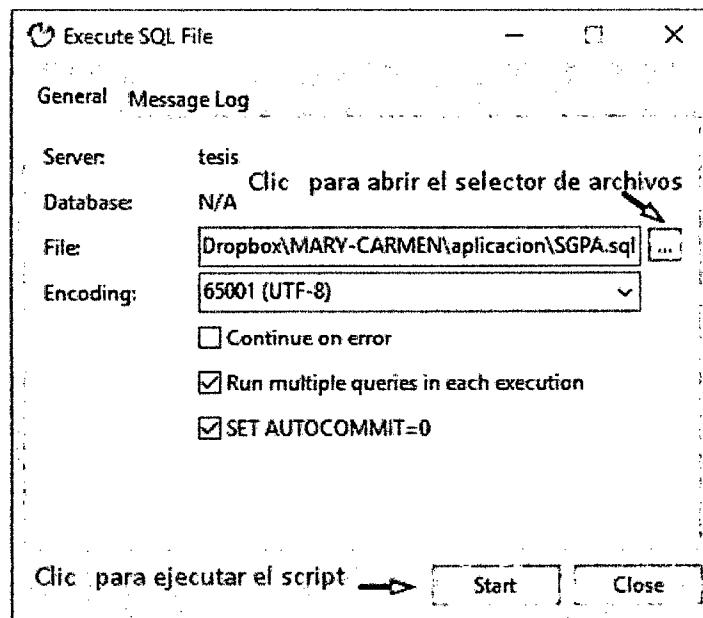
**Figura 2** Crear conexión

*Nota: Fuente propia*



5. Crear el esquema de base de datos y cargar los datos.

Para realizar esta tarea es necesario presionar anti clic en la conexión creada y seleccionar la opción "Execute SQL File". Una vez seleccionada esta opción aparecerá una ventana como se muestra en la Figura 3 donde será necesario seleccionar el script SGPA proporcionado al Secretario Académico.



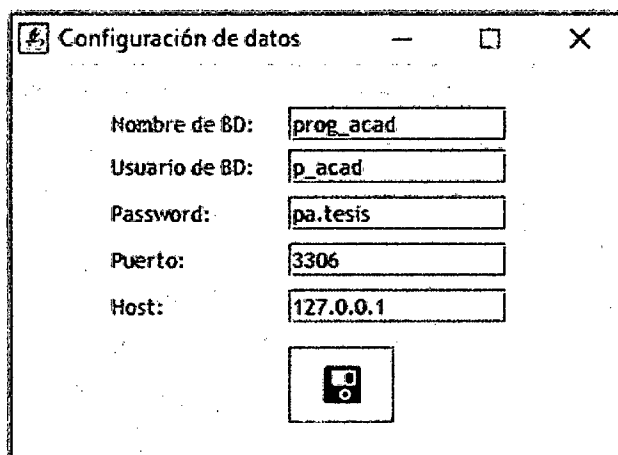
**Figura 3 Execute SQL File**

*Nota: Fuente propia*

## 2. MANUAL DE USUARIO

### 2.1 INGRESO AL SISTEMA

Para acceder a la aplicación, debe dar doble click en el ejecutable SGPA. Existe una configuración de acceso por defecto que permite acceder a la aplicación y en caso se determine que no se puede establecer una conexión con las credenciales guardadas se mostrará una ventana para que pueda proporcionarlas. Los datos necesarios se pueden apreciar en la Figura 4

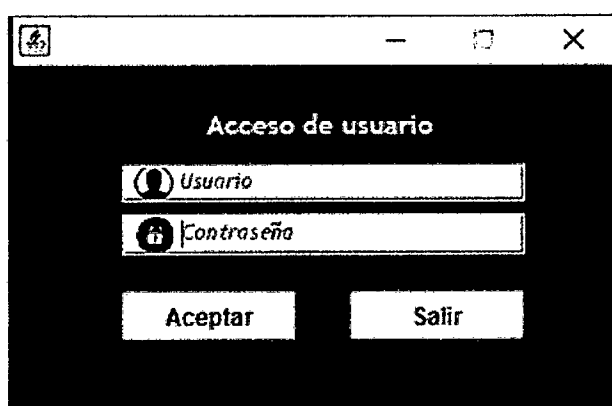


The screenshot shows a window titled "Configuración de datos" with a standard Windows-style title bar (minimize, maximize, close buttons). Inside the window, there are five labeled text input fields arranged vertically: "Nombre de BD:" with the value "prog\_acad", "Usuario de BD:" with the value "p\_acad", "Password:" with the value "pa.tesis", "Puerto:" with the value "3306", and "Host:" with the value "127.0.0.1". Below these fields is a button with a floppy disk icon, representing a "Guardar" (Save) function.

*Figura 4 Configuración de datos de acceso*

*Nota: Fuente propia*

Después de haber comprobado la conexión se mostrará el formulario de acceso al sistema donde debe introducir su nombre de usuario y contraseña respectivamente, luego presione el botón "Aceptar".



The screenshot shows a window titled "Acceso de usuario" with a dark background. It features two text input fields: the first is labeled "Usuario" with a person icon, and the second is labeled "Contraseña" with a lock icon. Below these fields are two buttons: "Aceptar" (Accept) and "Salir" (Exit).

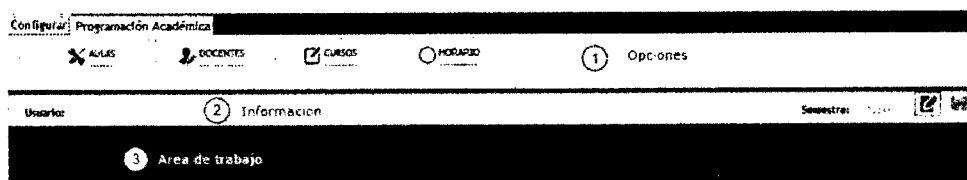
*Figura 5 Acceso de usuario*

*Nota: Fuente propia*

## 2.2 FUNCIONALIDAD GENERAL

La interfaz principal del sistema se divide en 3 zonas:

- Opciones (zona 1): Aquí encontraremos las diferentes opciones de acceso.
- Información (zona 2): Aquí se muestra información adicional al usuario, en la parte izquierda su nombre y a la derecha el semestre en el que se encuentra para generar la programación académica.
- Área de trabajo (zona 3): En esta zona se mostrarán las interfaces de las diferentes opciones a las que el usuario puede acceder.




*Figura 6. Zonas de la aplicación*

*Nota: Fuente propia*

### Botones e iconos

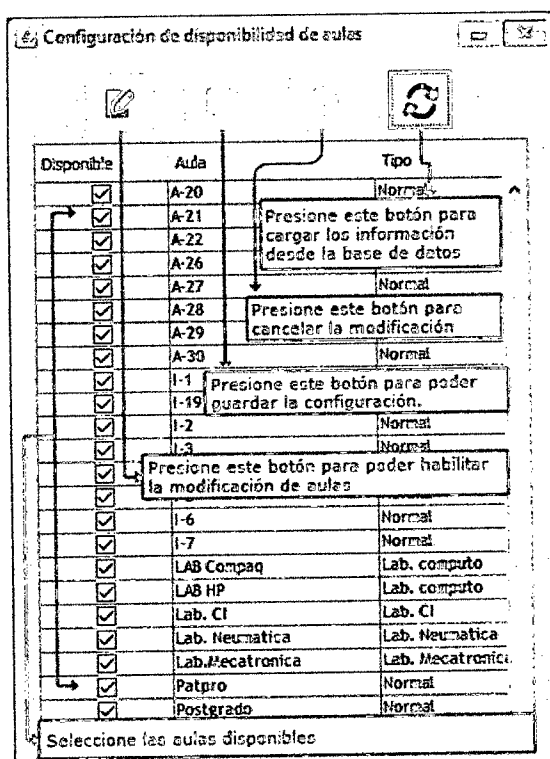
	Icono guardar: Permite guardar o almacenar información
	Icono editar: Permite habilitar los campos a modificar
	Icono cancelar: Permite cancelar la operación que se está realizando
	Icono cargar: Permite volver a cargar cierta información
	Icono buscar: Permite realizar consultas
	Icono limpiar: Permite limpiar cierta área de trabajo
	Icono carpeta: Permite cambiar la ruta donde se almacenarán los reportes
	Icono siguiente: Permite mover a la derecha los datos seleccionados
	Icono atrás: Permite mover a la izquierda los datos seleccionados
	Icono ejecutar: Permite ejecutar el algoritmo para generar la PA
	Icono detener: Permite pausar la ejecución del algoritmo
Configurar	Icono configurar: Permite agregar los cursos para su configuración
Agregar Escuela	Icono agregar escuela: Permite agregar una escuela a un curso seleccionado

Copiar curso	Icono copiar curso: Permite copiar un curso en la tabla de configuración
Eliminar curso	Icono eliminar curso: Permite eliminar un curso en la tabla de configuración
Habilitar para configurar	Icono habilitar para configurar Permite habilitar el check de asignación
Editar preferencias	Icono editar preferencias Permite editar las preferencias en horarios y cursos a los docentes
Más cursos	Icono más cursos: Permite mostrar otra ventana para buscar cursos de otros departamentos académicos.
	Icono gráfico: Permite visualizar el gráfico de aptitud de la solución en cada una de las generaciones.

## 2.3 FUNCIONALIDADES

### 2.3.1 Configurar disponibilidad de aulas

Para realizar la configuración de disponibilidad de aulas primero debe acceder a la opción “Aulas” del menú de configuración. En el formulario usted podrá modificar la disponibilidad de aulas, para ello debe hacer clic en el icono “Editar” y luego de realizar los cambios necesarios, presione el botón “Guardar”.



Disponible	Aula	Tipo
<input checked="" type="checkbox"/>	A-20	Normal
<input checked="" type="checkbox"/>	A-21	Normal
<input checked="" type="checkbox"/>	A-22	Normal
<input checked="" type="checkbox"/>	A-26	Normal
<input checked="" type="checkbox"/>	A-27	Normal
<input checked="" type="checkbox"/>	A-28	Normal
<input checked="" type="checkbox"/>	A-29	Normal
<input checked="" type="checkbox"/>	A-30	Normal
<input checked="" type="checkbox"/>	I-1	Normal
<input checked="" type="checkbox"/>	I-19	Normal
<input checked="" type="checkbox"/>	I-2	Normal
<input checked="" type="checkbox"/>	I-3	Normal
<input checked="" type="checkbox"/>	I-6	Normal
<input checked="" type="checkbox"/>	I-7	Normal
<input checked="" type="checkbox"/>	LAB Compaq	Lab. computo
<input checked="" type="checkbox"/>	LAB HP	Lab. computo
<input checked="" type="checkbox"/>	Lab. CI	Lab. CI
<input checked="" type="checkbox"/>	Lab. Neumatica	Lab. Neumatica
<input checked="" type="checkbox"/>	Lab. Mecatronica	Lab. Mecatronica
<input checked="" type="checkbox"/>	Patpro	Normal
<input checked="" type="checkbox"/>	Postgrado	Normal

Seleccione las aulas disponibles

**Figura 7. Configuración de disponibilidad de aulas**

*Nota: Fuente propia*

### 2.3.2 Configurar asignación docente

Para realizar la configuración de asignación de los docentes primero debe acceder a la opción “Docentes” del menú de configuración.

Una vez ubicado en el formulario podrá modificar la asignación de los docentes y para ello primero debe seleccionar el departamento académico y presionar el botón “Buscar” donde automáticamente el sistema presentará en la tabla la lista de los docentes que pertenecen al departamento seleccionado.

Para realizar alguna modificación debe hacer clic en el botón “habilitar para configurar” y luego de haber realizado los cambios necesarios, presione el botón “Guardar”. Si desea cancelar la modificación presione el botón “cancelar”.

Listado de docentes por departamento

Departamento Académico: **IO - INVESTIGACION DE OPERACIONES**

Asigna: Presione aquí para buscar los docentes del departamento seleccionado

Presione aquí para cancelar la modificación

Presione aquí para guardar cambios

Presione aquí para habilitar la edición de la casilla "Asignar"

Despliegue aquí para mostrar los diferentes departamentos

			Categoría
<input checked="" type="checkbox"/>	NESTOR MANUEL CASTILLO BURGOS	USIVA	PRINCIPAL
<input checked="" type="checkbox"/>	TEOBALDO	USIVA	PRINCIPAL
<input checked="" type="checkbox"/>	JULIO C	USIVA	PRINCIPAL
<input checked="" type="checkbox"/>	CARLOS	USIVA	PRINCIPAL
<input checked="" type="checkbox"/>	CARMEN GUTIERREZ	COMPLETO	ASOCIADO
<input checked="" type="checkbox"/>	LUCIANA MERCEDES TO		
<input type="checkbox"/>	NESTOR YOEL ALBAN P		
<input type="checkbox"/>	OLIVER FABIAN CUPEN		
<input type="checkbox"/>	ENRIQUE CRISANTO PALACIOS	POR DEFINIR	POR DEFINIR
<input type="checkbox"/>	ALDO SEGISMUNDO PEREDA CASTILLO	COMPLETO	EVENTUAL
<input type="checkbox"/>	HECTOR DANIEL QUINTANA RUIDIAS	POR DEFINIR	EVENTUAL
<input type="checkbox"/>	JULIAN DIESNTMAR LEON	POR DEFINIR	POR DEFINIR
<input type="checkbox"/>	GUILLERMO ALBERTO GARCIA MOGOLLOMA	COMPLETO	POR DEFINIR

Seleccione los docentes que se tendrán en cuenta para la asignación de clases

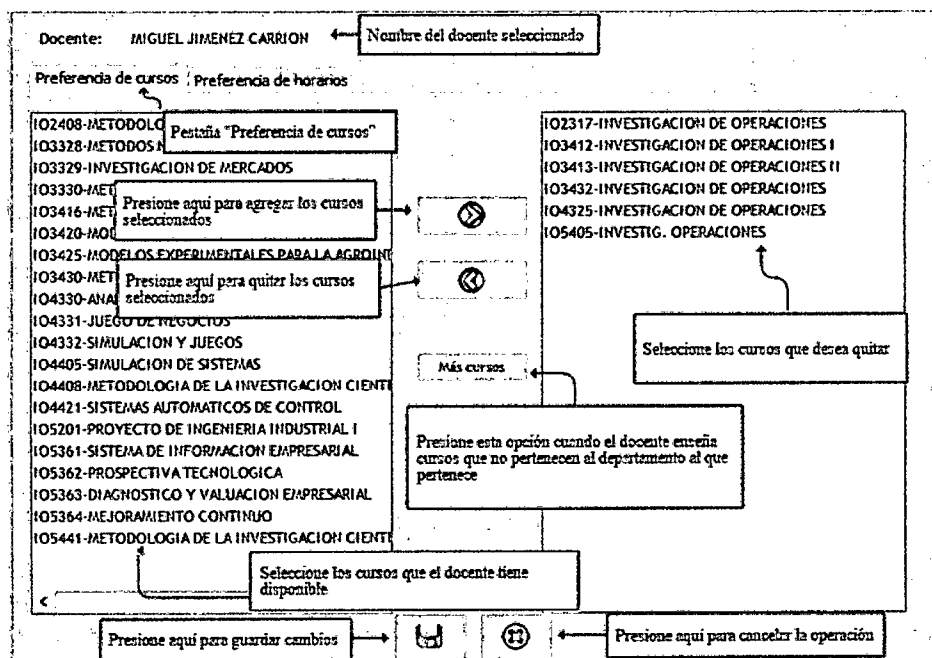
Figura 8 Configurar asignación docente

Nota: Fuente propia

### 2.3.3 Configurar preferencia de dictado de cursos

Cuando haya accedido a la venta de asignación docente explicada anteriormente y desee realizar la configuración de preferencia de dictado de cursos del docente usted debe seleccionar dicho docente dando doble clic sobre el para que se habilite la opción “Editar preferencias”. Una vez habilitada la opción debe dar clic y en la venta del costado debe ubicarse en la pestaña que dice “Preferencia de cursos”. Luego debe seleccionar los cursos de preferencia y presionar el botón “Siguiente”, en caso quiera quitar los cursos debe seleccionarlos y presionar el botón “Atrás”, después de haber agregado los cursos de preferencia debe hacer clic

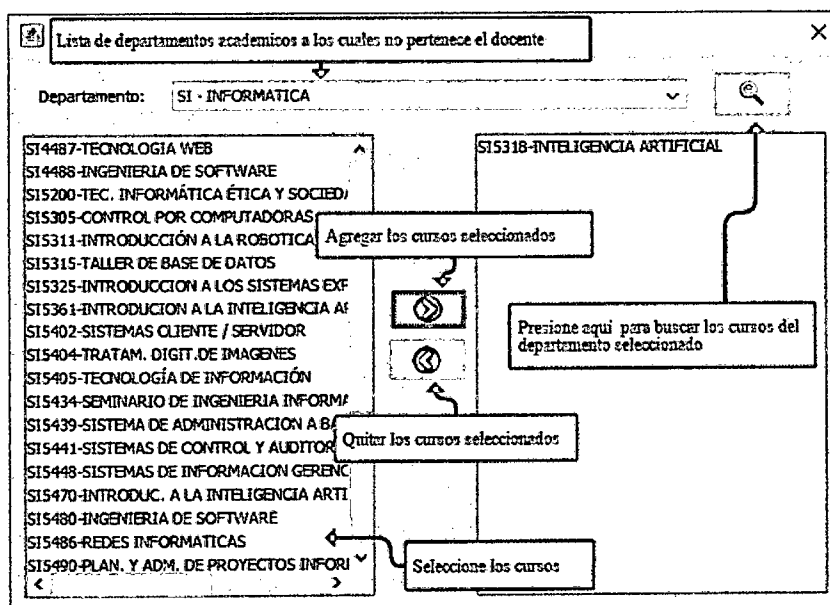
en el botón “Guardar”, en caso desista de la operación debe dar clic en la opción “Cancelar”.



**Figura 9 Configurar preferencia de dictado de cursos**

*Nota: Fuente propia*

Quando usted quiere configurar cursos a un docente que no sean del departamento al que este suscrito entonces debe de dar clic en la opción “Más cursos” el cual desplegará una nueva ventana para que pueda buscar los cursos que pertenecen a otros departamentos.



**Figura 10 Opción más cursos**

*Nota: Fuente propia*

### 2.3.4 Configurar disponibilidad de horarios

Cuando haya dado clic en la opción “Editar preferencias” debe ubicarse en la pestaña que dice “Preferencia de horarios”. Luego debe seleccionar los grupos de preferencia para cada uno de los docentes, para guardar la información presione el botón “Guardar” o el botón “Cancelar” en caso desista de la operación.

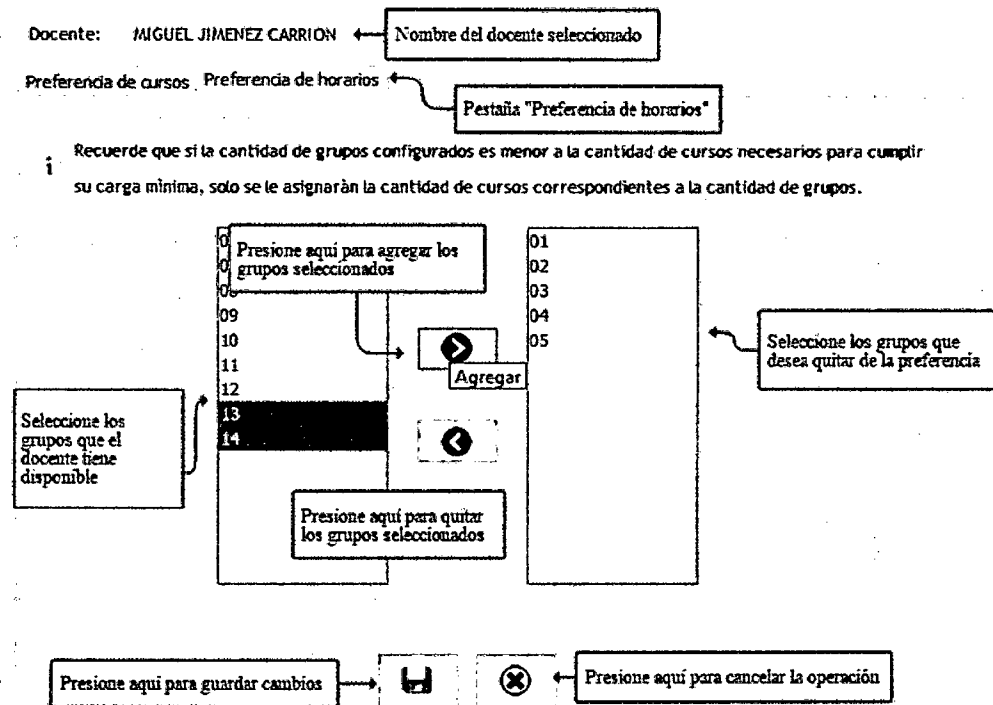


Figura 11 Configurar disponibilidad de horarios

Nota: Fuente propia

### 2.3.5 Configurar programación de cursos

Para realizar la configuración de cursos que se dictarán es necesario que primero acceda a la opción “Cursos” del menú de configuración. Cuando ya se encuentre allí debe de buscar los cursos dependiendo de la escuela y plan de estudios. Para realizar la búsqueda usted debe seleccionar el icono “Buscar”.

Luego usted tiene la opción de cambiar el número de secciones que se abrirán por curso, el tipo de aula y las escuelas para las que se dictaran dichos cursos. Una vez finalizada la configuración usted debe presionar en el botón “Guardar”





### 2.3.7 Configurar parámetros

Para que pueda realizar la modificación de los parámetros primero debe acceder la opción “Configurar” del menú programación académica.

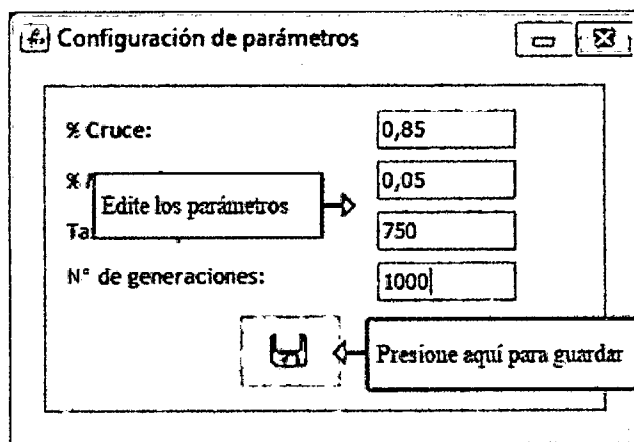


Figura 14 Configuración de parámetros

Nota: Fuente propia

### 2.3.8 Generar programación académica

Para poder generar la programación académica usted primero debe acceder a la opción “Programación académica”. Luego presione el botón “Generar” para que empiece el proceso de generación, si desea parar el proceso presione “Detener”.

En esta pantalla se muestran más opciones, cada una de ellas es explicada en la Figura 15.

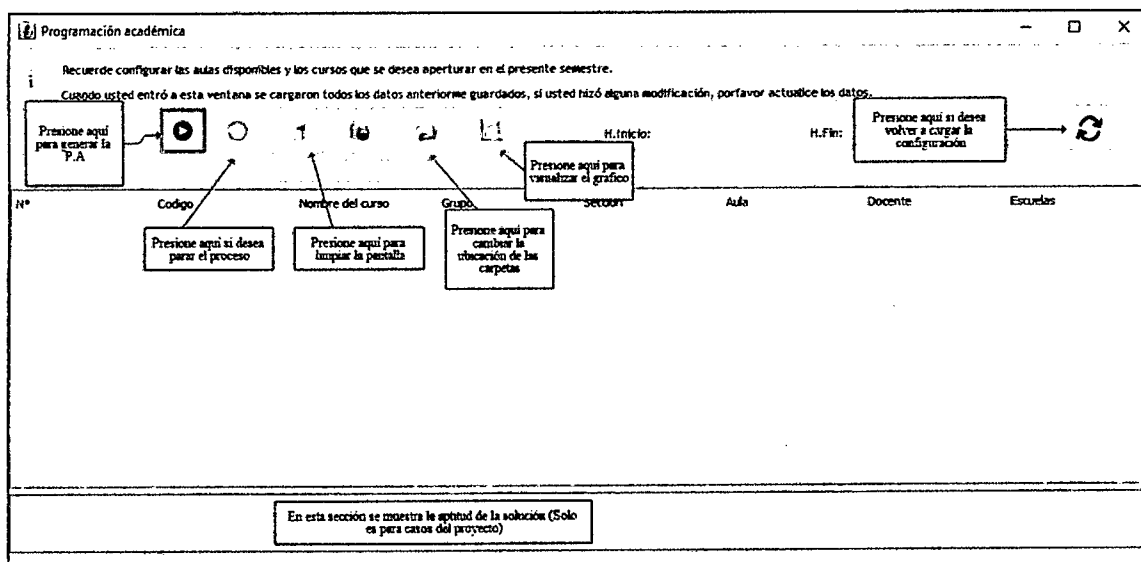


Figura 15 Generar programación académica

Nota: Fuente propia